

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	152	(garbage adj1 collect\$4) and cookie	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:34
L2	139	L1 and ((quer\$4 search\$4 retriev\$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:34
L3	23	L2 and (method near5 (handler bundle metadata))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:34
L4	13	L3 and ((run\$1time or (run adj1 time)) adj1 environment)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:34
L5	2	L4 and (method adj1 metadata)	US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/10/12 17:34
L6	12	(method adj1 metadata) and (garbage adj1 collect\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:35
L7	2	L6 and cookie	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:35
L8	0	2 and "N=2X"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:36
L9	1	2 and N-aligned	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:36

EAST Search History

L10	1	1 and N-aligned	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:37
L11	5702	707/100.ccls:	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:37
L12	0	11 and N-aligned	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:37
S1	4	"6317869".pn. "6671707".pn.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2007/10/12 17:33
S2	112	(garbage adj1 collect\$4) and cookie	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:12
S3	2	S2 and ((quer\$4 search\$4 retriev\$4) near5 (method adj1 metadata))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:20
S4	103	S2 and ((quer\$4 search\$4 retriev\$4))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:20
S5	20	S4 and (method near5 (handler bundle metadata))	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:21
S6	12	S5 and ((run\$1time or (run adj1 time)) adj1 environment)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:35
S7	2	S6 and (method adj1 metadata)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:36

EAST Search History

S8	9	(method adj1 metadata) and (garbage adj1 collect\$5)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:37
S9	2	S8 and cookie	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2007/10/12 17:35
S10	9	S8 and (access\$4 quer\$4 retriev\$4 receiv\$4)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	ON	2006/07/21 11:37
S11	161	(method adj1 metadata)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/21 12:32
S12	1	S11 near (method adj1 handle)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/21 12:28
S13	161	"method metadata"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/21 12:32
S14	3	S13 and "garbage collector"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/21 12:32
S15	4	(\$2direction\$4 adj1 search) near ((forward backward) adj1 search)	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/22 11:49
S16	2	(cach\$4 near (XML adj1 (metadata or data or (meta adj1 data)))) and cookie	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/12/10 22:34

Storing method metadata in code - Patent 20050177596

The article of claim 18, further comprising instructions that if executed enable the system to store the method **metadata** at an **N-aligned address** of the code ...

www.freepatentsonline.com/20050177596.html - 39k - [Cached](#) - [Similar pages](#) - [Note this](#)

[PDF] How to Shadow Every Byte of Memory Used by a Program

File Format: PDF/Adobe Acrobat - [View as HTML](#)

load/store, and Memcheck can simply ignore threading issues. On ... **address** is **N-aligned**, then sets N bytes at a time, and finally sets ...

valgrind.org/docs/shadow-memory2007.pdf - [Similar pages](#) - [Note this](#)

[PDF] Dynamic Binary Analysis and Instrumentation Nicholas Nethercote

File Format: PDF/Adobe Acrobat

Tools can only see FP and SIMD load/store **addresses** and data widths. Fischer [86] also **store metadata** separately, in order to perform the checking ...

valgrind.org/docs/phd2004.pdf - [Similar pages](#) - [Note this](#)

[PDF] How to Shadow Every Byte of Memory Used by a Program

registers, nor the heap block **metadata**, as they have been covered **address** is **N-aligned**, then sets N bytes at a time, and finally sets ...

portal.acm.org/ft_gateway.cfm?id=1254820&type=pdf&coll=&dl=ACM&CFID=15151515&CFTOKEN=... - [Similar pages](#) - [Note this](#)

[PDF] Dynamic binary analysis and instrumentation

File Format: PDF/Adobe Acrobat

for tools that **store address** traces for processing off-line. Patil and Fischer [86] also **store metadata** separately, in order to ...

www.cl.cam.ac.uk/techreports/UCAM-CL-TR-606.pdf - [Similar pages](#) - [Note this](#)

Information Processing & Management : Web page title extraction ...

In this paper, we take a supervised machine learning approach to **address** the problem ...

Keywords: Information retrieval; HTML document; **Metadata** extraction ...

linkinghub.elsevier.com/retrieve/pii/S0306457306001981 - [Similar pages](#) - [Note this](#)

[PDF] MonetDB Reference Manual

File Format: PDF/Adobe Acrobat

XML is often used to **store** annotations (i.e. **meta-data**, data describing other with IPv4 **addresses** and URLs using inet, url, and several types to ...

monetdb.cwi.nl/projects/monetdb/Assets/manual-2x1.pdf - [Similar pages](#) - [Note this](#)

[PDF] MonetDB Server Reference Manual

File Format: PDF/Adobe Acrobat

address CMDgetBATlrefcnt comment "Utility for debugging MAL interpreter"; 8.5 Constants. The const module provides a box abstraction **store** for global ...

monetdb.cwi.nl/projects/monetdb//Assets/m5manual-2x1.pdf - [Similar pages](#) - [Note this](#)

[PDF] Monitoring sites in New Zealand and an analysis of site ...

File Format: PDF/Adobe Acrobat - [View as HTML](#)

34 monitoring sites were included in the collation of **metadata** for this The monitor is located within a shop **store** at the corner of Vivian St and ...

www.mfe.govt.nz/publications/air/air-quality-site-monitoring-jul98.pdf -

[Similar pages](#) - [Note this](#)

Summary of changes from v2.6.8 to v2.6.9 ...

This patch fixes it by creating a temporary to **store** the register whose **address** we want to get and stuff into the per-bte info structure. ...
www.kernel.org/pub/linux/kernel/v2.6/ChangeLog-2.6.9 - 977k -
Cached - Similar pages - Note this

*In order to show you the most relevant results, we have omitted some entries very similar to the 10 already displayed.
If you like, you can repeat the search with the omitted results included.*

Try [Google Desktop](#): search your computer as easily as you search the web.

N-aligned addresses store metadata

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

©2007 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

[PDF] How to Shadow Every Byte of Memory Used by a ProgramFile Format: PDF/Adobe Acrobat - [View as HTML](#)**address** is **N-aligned**, then sets **N** bytes at a time, and finally sets values are never read—but we need to **garbage collect** (GC) the ...valgrind.org/docs/shadow-memory2007.pdf - [Similar pages](#) - [Note this](#)**[PDF] Dynamic Binary Analysis and Instrumentation Nicholas Nethercote**

File Format: PDF/Adobe Acrobat

Tools can only see FP and SIMD load/store **addresses** and data widths. Fischer [86]also **store metadata** separately, in order to perform the checking ...valgrind.org/docs/phd2004.pdf - [Similar pages](#) - [Note this](#)**[PDF] Dynamic binary analysis and instrumentation**

File Format: PDF/Adobe Acrobat

for tools that **store address** traces for processing off-line. Patil and Fischer [86] also**store metadata** separately, in order to ...www.cl.cam.ac.uk/techreports/UCAM-CL-TR-606.pdf - [Similar pages](#) - [Note this](#)**[PDF] MonetDB Reference Manual**

File Format: PDF/Adobe Acrobat

XML is often used to **store** annotations (i.e. **meta-data**, data describing other data).**address** CMDBATsetGarbage comment "Designate a BAT as **garbage**"; ...monetdb.cwi.nl/projects/monetdb/Assets/manual-2x1.pdf - [Similar pages](#) - [Note this](#)**[PDF] MonetDB Server Reference Manual**

File Format: PDF/Adobe Acrobat

are **garbage** collected before the function returns the result. **address** INSPECTatom

names comment "Collect a BAT with the atom names."; ...

monetdb.cwi.nl/projects/monetdb//Assets/m5manual-2x1.pdf - [Similar pages](#) - [Note this](#)**Summary of changes from v2.6.8 to v2.6.9 ...**This patch fixes it by creating a temporary to **store** the register whose **address** we want to get and stuff into the per-byte info structure. ...

www.kernel.org/pub/linux/kernel/v2.6/ChangeLog-2.6.9 - 977k -

[Cached](#) - [Similar pages](#) - [Note this](#)*In order to show you the most relevant results, we have omitted some entries very similar to the 6 already displayed.**If you like, you can [repeat the search with the omitted results included](#).*Try [Google Desktop](#): search your computer as easily as you search the web.

[Web](#) [Images](#) [Video](#) [News](#) [Maps](#) [Gmail](#) [more ▾](#)farhan.syed@gmail.com | [My Account](#) | [Sign out](#)[Google](#)

cookie N-aligned addresses store metadata cc

[Advanced Search](#)[Preferences](#)[New! View and manage your web history](#)[Web](#)Results 1 - 1 of 1 for cookie N-aligned addresses store metadata collect garbage (0.16 seconds)[\[PDF\] MonetDB Reference Manual](#)

File Format: PDF/Adobe Acrobat

XML is often used to **store annotations** (i.e. **meta-data**, data describing other data).**address CMDBATsetGarbage** comment "Designate a BAT as **garbage**"; ...monetdb.cwi.nl/projects/monetdb/Assets/manual-2x1.pdf - [Similar pages](#) - Note this[Download Google Pack](#): free essential software for your PCcookie N-aligned addresses store m [Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)©2007 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

Google

cookie aligned addresses store metadata colle

[Advanced Search](#)[Preferences](#)[New! View and manage your web history](#)

Web

Results 1 - 10 of about 29,400 for cookie aligned addresses store metadata collect garbage (0.32 seconds)**[PDF] Technologien für Verteilte Systeme (Oberleitner, Zdun, Rosenberg ...)**File Format: PDF/Adobe Acrobat - [View as HTML](#)They contain the class structure and additional **metadata** as well as the distribution list or multicast **addresses**. The persistent **store** can be an SQL ...josefine.ben.tuwien.ac.at/~mjaros/files/tvs3.pdf - [Similar pages](#) - Note this00:00:00 --- log: started forth/05.09.23 00:20:19 <Quartus> Ok ...I'm sooooo "pretty" ;-) 12:12:29 <tathi> Some platforms, like x86, allow you to use any **address**, but it's slower if you're not **aligned** ...tunes.org/~nef/logs/forth/05.09.23 - 126k - [Cached](#) - [Similar pages](#) - Note thismacgeekpro's bookmarks tagged with "store" on del.icio.usto shopping **store** iphone oem parts accessories ... saved by 18 other people ... to iphone webapp itunes music **store** concert event social community artist ...del.icio.us/macgeekpro/**store** - 197k - [Cached](#) - [Similar pages](#) - Note thisMailing list archives(2.0 + 1.3) PR 26002. modules/**metadata**/mod_usertrack.c: r1.44 +1: nd * mod_usertrack:Use always the **Cookie** header to determine the **cookie** name (2.0 + 1.3). ...

mail-archives.apache.org/mod_mbox/httpd-dev/200401.mbox/%

3C200401150445.i0F4jBm12388@Boron.MeepZor.Com%3E - 55k -

[Cached](#) - [Similar pages](#) - Note thisMailing list archives(2.0 + 1.3) PR 26002. modules/**metadata**/mod_usertrack.c: r1.44 +1: nd, bnicholes, trawick* mod_usertrack: Use always the **Cookie** header to determine the ...

mail-archives.apache.org/mod_mbox/httpd-dev/200401.mbox/%

3C200401290445.i0T4jB130514@Boron.MeepZor.Com%3E - 55k -

[Cached](#) - [Similar pages](#) - Note this

[More results from mail-archives.apache.org]

System and method for maintaining consistent independent server ...While the foregoing discussion **addresses** session state maintenance, To prevent overflow of the **cookie**, some embodiments will **garbage-collect** old ...

www.patentstorm.us/patents/7024451-description.html - 156k -

[Cached](#) - [Similar pages](#) - Note this**[PDF] IBM WebSphere Portal V4.1 Handbook Volume 3**File Format: PDF/Adobe Acrobat - [View as HTML](#)the Delete button and a blue **garbage** can will appear beside it, as shown in The administrative database is used to **store metadata** and other ...www.redbooks.ibm.com/redbooks/pdfs/sg246921.pdf - [Similar pages](#) - Note thisMagic data for file(1) command. # Format is described in magic(5 ...Hash 1.85/1.86 databases **store metadata** in network byte order. Kleff) 10 string MIT-

MAGIC-COOKIE-1 X11 Xauthority data 11 string MIT-MAGIC-COOKIE-1 X11 ...

svn.gnu-monks.org/trunk/reveng-tools/magic_ofs/magic - 176k -

[Cached](#) - [Similar pages](#) - Note thistmp.dataJava doesn't keep the process around // as it **stores metadata** in a Process object.canonicalHostName + '=>' + **addresses.collect{ it. ...**pleac.sourceforge.net/pleac_groovy.html - 623k - [Cached](#) - [Similar pages](#) - Note this

O'Reilly Media -- Bookstore: Amazon Hacks

The wish list also **stores** your **address** so your aunt won't have to try to remember
MP3s **store metadata** in ID3 tags (at the end of the MP3 file) that ...
www.oreilly.com/catalog/amazonhks/toc.html - 591k - Cached - Similar pages - Note this

1 2 3 4 5 6 7 8 9 10 [**Next**](#)

Download [Google Pack](#): free essential software for your PC

[Search](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#) | [Dissatisfied? Help us improve](#)

©2007 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

Google

cookie aligned addresses store metadata colle Advanced Search
 New! View and manage your web history

Web

Results 11 - 20 of about 29,400 for cookie aligned addresses store metadata collect garbage (0.08 seconds)[PDF] P15155 Security White PaperFile Format: PDF/Adobe Acrobat - View as HTML

budgets may not appropriately align to address application risks in a manner Security intelligence for applications is meant to collect, analyze and ...

www-935.ibm.com/services/us/imc/pdf/wp-p15155-security-white-paper.pdf -

Similar pages - Note this

System and method for maintaining consistent independent server ...

Next, it updates its local copy of the user's email address based on the To prevent overflow of the cookie, some embodiments will garbage-collect old ...

www.freepatentsonline.com/20030105805.html - 163k - Cached - Similar pages - Note this

System and method for maintaining consistent independent server ...

Specifically, when the Web server desires to store state data client-side, embodiments will garbage-collect old Event Descriptors in the cookie at ...

www.freepatentsonline.com/7024451.html - 181k - Cached - Similar pages - Note this

Interview_Faq

What is the difference between **Metadata** and **Manifest** ? How do you enforce **garbage collection** in .NET? System.GC.Collect(); Can you declare a C++ type ...

www.scribd.com/doc/72848/InterviewFaq - 143k - Cached - Similar pages - Note this

[PDF] Automatic Web PublishingFile Format: PDF/Adobe Acrobat - View as HTML

A **cookie** is introduced to the client by including a Set-Cookie header which is basically a text file with **metadata** associated, and **stores** it in the ...

<https://www.cs.tcd.ie/publications/tech-reports/reports.00/TCD-CS-2000-55.pdf> -

Similar pages - Note this

[PDF] Securing history: Privacy and accountability in database systemsFile Format: PDF/Adobe Acrobat - View as HTML

In Firefox 2.0, remote. sites can **store** data that persists across sessions in an SQLite database as a sophisticated replacement of **cookies** ...

www-db.cs.wisc.edu/cidr/cidr2007/papers/cidr07p44.pdf - Similar pages - Note this

Differences between mono-1.1.2.20041102/mono/metadata/reflection.c ...

127, * tracked by the **garbage** collector. We use libgc because it's 5422, /* Note: the cache is based on the **address** of the signature into the method ...

dsrg.mff.cuni.cz/.../mono/diffs/showdiff.php?old=2004-11-02&new=2004-11-03&diffIndex=2&bench= - Similar pages - Note this

HTML 5

Resource **metadata** management. The URL attribute must return the document's **address**.

The domain attribute must be initialised to the document's domain, ...

www.whatwg.org/specs/web-apps/current-work/ - 977k - Cached - Similar pages - Note this

ASP.NET News

Since Python 2.2 there is a handy function in the **Garbage** Collection Module as an email sink, **collecting** all received emails destined for any **address** ...

aspn.activestate.com/ASP.NET/News/articles_html?resource_id=68579&num=200 - 75k - Cached - Similar pages - Note this

[PDF] RScheme

File Format: PDF/Adobe Acrobat

<http://www.google.com/search?q=cookie+aligned+addresses+store+metadata+collect+garbage&hl=en&s...> 10/12/2007

Another way of putting this is that **addresses** should be **aligned** on a four-byte In order for it to be possible to **garbage collect** in the presence of ...
www.rscheme.org/rs/b/0.7.3.4/5/rscheme.pdf - [Similar pages](#) - [Note this](#)

[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [Next](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#)

©2007 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

Google

cookie aligned addresses store metadata colle Advanced Search
Preferences
New! View and manage your web history

Web

Results 21 - 30 of about 29,400 for cookie aligned addresses store metadata collect garbage. (0.13 seconds)WGA Developer Center

Bugfix: Hibernate SessionFactoryImpl and child objects are not garbage collected WGA needs at some places the ability to store "metadata information" ...
dev.innovationgate.com/.../html/default/wsci-5lgqdp.en.html;jsessionid=A62B4EE6F6BF3E8D5FBF9C7B5414FE5 - 485k -
 Cached - Similar pages - Note this

LCOV - aggregate tests - dyninstAPI/src/process.C

This is done 505 : // for two reasons: first, the symbol **address** might be off 5824 : }
 5825 : 5826 : // **garbage collect** instrumentation 5827 : void ...
www.dyninst.org/coverage/dyninstAPI/src/process.C.gcov.html - 611k -
 Cached - Similar pages - Note this

The Book of Visual Studio NET - A Guide for Developers

The GUID value that the **cookie** maintains is related to data that is stored in Providing the ability to **store all metadata** within the assembly has an ...
www.scribd.com/doc/7473/The-Book-of-Visual-Studio-NET-A-Guide-for-Developers -
 Similar pages - Note this

[PDF] F T P C S

File Format: PDF/Adobe Acrobat - [View as HTML](#)
 The **metadata** server **stores** each file's striping and clustering information in To free space, the system must **garbage-collect** dead data, a process ...
www.cs.princeton.edu/~ras/thesisfinal.pdf - Similar pages - Note this

IBM - Fix List for Lotus Notes and Lotus Domino Release 6.5.6 ...

The cause of the panic was because an **address** was passed that should have been a by setting it to NULL, which allows the JVM to **garbage collect** it. ...
www.ibm.com/support/docview.wss?rs=3025&context=SSPQ69&dc=DA400&uid=swg27009492&loc=en_US... - 355k -
 Cached - Similar pages - Note this

Ejohnson - Eric Johnson's Personal website > Home

This month: cleaning up resources that are not **garbage** collected by the CLR NET platform uses **metadata** and assemblies to **store** information about ...
www.ejohnson.ws/ - 686k - Cached - Similar pages - Note this

Scraping <http://www.nitobi.com...> Adding stuff.. done! Title ...

While Nitobi may use **cookies** to track your visit to our website, and our Web servers automatically log the IP/Internet **address** of your computer, ...
www.ebusiness-apps.com/admin/scrapesite.asp - 194k - Cached - Similar pages - Note this

[PDF] A DIGITAL LIBRARY COMPONENT ASSEMBLY ENVIRONMENT

File Format: PDF/Adobe Acrobat - [View as HTML](#)
 Archive that **stores metadata** records as XML or text files displayed as a link to the corresponding URL, **aligned** from left to right at the top of the ...
pubs.cs.uct.ac.za/archive/00000306/01/csc500_2005_linda_eyambe_thesis.pdf -
 Similar pages - Note this

Data processing: database and file management or data structures ...

Stored individual slide files are populated with **metadata** for allowing subsequent
 20060294166 - Arrangement and method for **garbage** collection in a ...
www.freshpatents.com/Data-processing--database-and-file-management-or-data-structures-dt200612ntc707.php - 249k - Cached - Similar pages - Note this

[PDF] [SWI-Prolog and the Web](#)

File Format: PDF/Adobe Acrobat - [View as HTML](#)

achievable in a 32-bit **address**-space we decided to implement the low level **store**. in C.

Profiting from the known uniform structure of the data we realised ...

hcs.science.uva.nl/projects/SWI-Prolog/articles/TPLP-plweb.pdf - [Similar pages](#) - [Note this](#)

[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [Next](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#)

©2007 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)

Google

cookie aligned addresses store metadata colle

[Advanced Search](#)[Preferences](#)[New! View and manage your web history](#)

Web

Results 31 - 40 of about 29,400 for cookie aligned addresses store metadata collect garbage (0.11 seconds)Magic # Magic data for file(1) command. # Machine-generated from ...Hash 1.85/1.86 databases **store metadata** in network byte order. modern format, 8-bit

aligned 0 string zz MGR bitmap, old format, 1-bit deep, ...

www.nsrl.nist.gov/documents/hashapalooza2004/slides/magic.txt - 278k -

[Cached](#) - [Similar pages](#) - [Note this](#)LIST OF CHANGES FROM PREVIOUS RELEASES: <\$Revision: 1.9 \$> Changes ...Make m68k ports use them consistently, **garbage-collecting** redundant copies. Leaffs's now **store** the **address** of their struct lock in v_vnlock. ...

ftp.netbsd.org/pub/NetBSD/NetBSD-2.0.3/CHANGES.prev - 366k -

[Cached](#) - [Similar pages](#) - [Note this](#)TechbloggingThere is a couple of package **metadata** we currently are not tracking inside askdjfhaskdfOR -askdjfhaskdf - top result for this **garbage** word: Google! ...blog.drinsama.de/erich/en/xml - 93k - [Cached](#) - [Similar pages](#) - [Note this](#)WSA Software > HomeNET Framework, **addresses** this need. Using the WS-Security framework, NETplatform uses **metadata** and assemblies to **store** information about components, ...www.wsasoft.com/ - 650k - [Cached](#) - [Similar pages](#) - [Note this](#)NEOHAPSIS - Peace of Mind Through Integrity and InsightReferer, and **Cookie** headers, Web forwarder (accelerator) mode, backgrounddownloading, multiple logs, fast operation, very configurable **garbage** ...

archives.neohapsis.com/archives/apps/freshmeat/2002-11/0023.html - 111k -

[Cached](#) - [Similar pages](#) - [Note this](#)Password class. Cug Class using an array. extended from Bs_Cug ...Collect the results of running a set of TestCases. Constructor. eg to **store** in the database. Validates the syntax of a given email **address** ...developer.blueshoes.org/phpdoc/elementlist.xml - 377k - [Cached](#) - [Similar pages](#) - [Note this](#)[trace] Log of /nixHandle weird cases when the server redirects us while setting a **cookie**. TODO:acquire a global lock on the **store** while **garbage collecting** ...

https://svn.cs.uu.nl:12443/viewvc/trace/nix/?view=log - 824k -

[Cached](#) - [Similar pages](#) - [Note this](#)Software Carpentry (Version 1135)Packing with **Metadata**. First step is to **store** a list of identically-structured records to aWhen its count drops to 0, Python can **garbage collect** it ...swc.scipy.org/lec/unified.html - 977k - [Cached](#) - [Similar pages](#) - [Note this](#)[PDF] DESIGN AND IMPLEMENTATION OF HIGH PERFORMANCE FILE SYSTEMSFile Format: PDF/Adobe Acrobat - [View as HTML](#)Traditional file systems use simple offsets as directory **cookies**, to **store metadata** as well as file data. The log partition is organized in the ...www.cs.binghamton.edu/~zzhang/zhang8-7.pdf - [Similar pages](#) - [Note this](#)Livelock -- concepts Interface interrupts used to schedule network ...XDR defines: size, byte-order, **alignment** of basic data types DEFINE args & results a **garbage** file is left on the server - also access perms on file can ...

www.stanford.edu/~stinson/cs240/final_revs/revs2.txt - 87k -

[Cached](#) - [Similar pages](#) - [Note this](#)

[Previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#) [Next](#)

[Search within results](#) | [Language Tools](#) | [Search Tips](#)

©2007 Google - [Google Home](#) - [Advertising Programs](#) - [Business Solutions](#) - [About Google](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
 The ACM Digital Library The Guide

cookie aligned addresses store metadata collect garbage

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used:

[cookie aligned addresses store metadata collect garbage](#)

Found 34,547 of 212,128

Sort results by relevance [Save results to a Binder](#)[Try an Advanced Search](#)Display results expanded form [Search Tips](#)[Try this search in The ACM Guide](#) [Open results in a new window](#)

Results 1 - 20 of 200

Result page: **1** [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

1 [Programming languages: Garbage collection for embedded systems](#)

David F. Bacon, Perry Cheng, David Grove

September 2004 **Proceedings of the 4th ACM international conference on Embedded software EMSOFT '04****Publisher:** ACM PressFull text available: [pdf\(199.59 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Security concerns on embedded devices like cellular phones make Java an extremely attractive technology for providing third-party and user-downloadable functionality. However, garbage collectors have typically required several times the maximum live data set size (which is the minimum possible heap size) in order to run well. In addition, the size of the virtual machine (ROM) image and the size of the collector's data structures (metadata) have not been a concern for server- or workstation-ori...

Keywords: compaction, fragmentation, mark-and-sweep, tracing**2** [Service-oriented and mobile computing: From representations to computations: the evolution of web architectures](#)

Justin R. Erenkrantz, Michael Gorlick, Girish Suryanarayana, Richard N. Taylor

September 2007 **Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering ESEC-FSE '07****Publisher:** ACM PressFull text available: [pdf\(416.31 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

REpresentational State Transfer (REST) guided the creation and expansion of the modern web. What began as an internet-scale distributed hypermedia system is now a vast sea of shared and interdependent services. However, despite the expressive power of REST, not all of its benefits are consistently realized by working systems. To resolve the dissonance between the promise of REST and the reality of fielded systems, we critically examine numerous web architectures. Our investigation yields a se ...

Keywords: mobile code, network continuations, representational state transfer, web services**3** [MC²: high-performance garbage collection for memory-constrained environments](#)

Narendran Sachindran, J. Eliot B. Moss, Emery D. Berger

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN**

conference on Object-oriented programming, systems, languages, and applications OOPSLA '04, Volume 39 Issue 10

Publisher: ACM Press

Full text available:  pdf(503.53 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory are essential. Typical collection techniques used on these devices are mark-sweep and mark-compact. Mark-sweep collectors can provide good throughput and pause times but suffer from fragmentation. Mark-compact collectors prevent fragmentation, ...

Keywords: copying collector, generational collector, java, mark-compact, mark-copy, mark-sweep, memory-constrained copying

4 Garbage collection without paging

 Matthew Hertz, Yi Feng, Emery D. Berger

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05, Volume 40 Issue 6**

Publisher: ACM Press

Full text available:  pdf(231.14 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Garbage collection offers numerous software engineering advantages, but interacts poorly with virtual memory managers. Existing garbage collectors require far more pages than the application's working set and touch pages without regard to which ones are in memory, especially during full-heap garbage collection. The resulting paging can cause throughput to plummet and pause times to spike up to seconds or even minutes. We present a garbage collector that avoids paging. This *bookmarking collect* ...

Keywords: bookmarking collection, garbage collection, generational collection, memory pressure, paging, virtual memory

5 Garbage collection: Reducing generational copy reserve overhead with fallback compaction

 Phil McGahey, Antony L. Hosking

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available:  pdf(817.15 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As programming languages with managed runtimes become increasingly popular, it is essential that virtual machines are implemented efficiently. The performance of the memory management subsystem can be a defining factor in the performance of the virtual machine as a whole. We present a technique by which garbage collector performance can be improved. We describe an algorithm that combines a standard generational copying collector with a mark and compact collector. We observe that, since most objects ...

Keywords: copying collector, garbage collection, generational collector, java, mark and compact

6 Virgil: objects on the head of a pin

 Ben L. Titzer

October 2006 **ACM SIGPLAN Notices , Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06, Volume 41 Issue 10**

Publisher: ACM Press

Full text available: [pdf\(487.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Embedded microcontrollers are becoming increasingly prolific, serving as the primary or auxiliary processor in products and research systems from microwaves to sensor networks. Microcontrollers represent perhaps the most severely resource-constrained embedded processors, often with as little as a few bytes of memory and a few kilobytes of code space. Language and compiler technology has so far been unable to bring the benefits of modern object-oriented languages to such processors. In this paper ...

Keywords: data-sensitive optimization, dead code elimination, embedded systems, heap compression, microcontrollers, multi-stage computation, sensor networks, standalone programs, static analysis, systems software, whole-program compilation

7 Mostly-copying reachability-based orthogonal persistence

 Antony L. Hosking, Jiawan Chen

October 1999 **ACM SIGPLAN Notices**, **Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available: [pdf\(3.25 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe how reachability-based orthogonal persistence can be supported even in uncooperative implementations of languages such as C++ and Modula-3, and without modification to the compiler. Our scheme extends Bartlett's mostly-copying garbage collector to manage both transient objects and resident persistent objects, and to compute the reachability closure necessary for stabilization of the persistent heap. It has been implemented in our prototype of reachability-based persistence for M ...

8 CCured: type-safe retrofitting of legacy software

 George C. Necula, Jeremy Condit, Matthew Harren, Scott McPeak, Westley Weimer
May 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 3

Publisher: ACM Press

Full text available: [pdf\(613.66 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article describes CCured, a program transformation system that adds type safety guarantees to existing C programs. CCured attempts to verify statically that memory errors cannot occur, and it inserts run-time checks where static verification is insufficient. CCured extends C's type system by separating pointer types according to their usage, and it uses a surprisingly simple type inference algorithm that is able to infer the appropriate pointer kinds for existing C programs. CCured uses phys ...

Keywords: Memory safety, libraries, pointer qualifier, subtyping

9 The Conquest file system: Better performance through a disk/persistent-RAM hybrid

 design

An-I Andy Wang, Geoff Kuenning, Peter Reiher, Gerald Popek
August 2006 **ACM Transactions on Storage (TOS)**, Volume 2 Issue 3

Publisher: ACM Press

Full text available: [pdf\(1.34 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Modern file systems assume the use of disk, a system-wide performance bottleneck for over a decade. Current disk caching and RAM file systems either impose high overhead to access memory content or fail to provide mechanisms to achieve data persistence across reboots. The Conquest file system is based on the observation that memory is becoming inexpensive, which enables all file system services to be delivered from memory, except for providing large storage capacity. Unlike caching, Con ...

Keywords: Persistent RAM, file systems, performance measurement, storage management

10 Software support for portable storage: A superblock-based flash translation layer for

NAND flash memory

 Jeong-Uk Kang, Heeseung Jo, Jin-Soo Kim, Joonwon Lee
October 2006 **Proceedings of the 6th ACM & IEEE International conference on Embedded software EMSOFT '06**

Publisher: ACM Press

Full text available: [pdf\(487.09 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In NAND flash-based storage systems, an intermediate software layer called a flash translation layer (FTL) is usually employed to hide the erase-before-write characteristics of NAND flash memory. This paper proposes a novel superblock-based FTL scheme, which combines a set of adjacent logical blocks into a superblock. In the proposed FTL scheme, superblocks are mapped at coarse granularity, while pages inside the superblock are mapped freely at fine granularity to any location in several physical b ...

Keywords: NAND flash memory, address translation, flash translation layer (FTL)

11 An efficient and backwards-compatible transformation to ensure memory safety of C

programs

 Wei Xu, Daniel C. DuVarney, R. Sekar
October 2004 **ACM SIGSOFT Software Engineering Notes, Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering SIGSOFT '04/FSE-12**, Volume 29 Issue 6

Publisher: ACM Press

Full text available: [pdf\(115.69 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Memory-related errors, such as buffer overflows and dangling pointers, remain one of the principal reasons for failures of C programs. As a result, a number of recent research efforts have focused on the problem of dynamic detection of memory errors in C programs. However, existing approaches suffer from one or more of the following problems: inability to detect all memory errors (e.g., Purify), requiring non-trivial modifications to existing C programs (e.g., Cyclone), changing the memory ma ...

Keywords: C, memory safety, program transformation

12 The KaffeOS Java runtime system

 Godmar Back, Wilson C. Hsieh
July 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: [pdf\(704.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Single-language runtime systems, in the form of Java virtual machines, are widely deployed platforms for executing untrusted mobile code. These runtimes provide some of the features that operating systems provide: interapplication memory protection and basic system services. They do not, however, provide the ability to isolate applications from each other. Neither do they provide the ability to limit the resource consumption of applications. Consequently, the performance of current systems degra ...

Keywords: Robustness, garbage collection, isolation, language runtimes, resource management, termination, virtual machines

13 Quantifying the performance of garbage collection vs. explicit memory management Matthew Hertz, Emery D. BergerOctober 2005 **ACM SIGPLAN Notices , Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '05**, Volume 40 Issue 10

Publisher: ACM Press

Full text available:  pdf(1.51 MB)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Garbage collection yields numerous software engineering benefits, but its quantitative impact on performance remains elusive. One can compare the cost of *conservative* garbage collection to explicit memory management in C/C++ programs by linking in an appropriate collector. This kind of direct comparison is not possible for languages designed for garbage collection (e.g., Java), because programs in these languages naturally do not contain calls to free. Thus, the actual gap between the tim ...

Keywords: explicit memory management, garbage collection, oracular memory management, paging, performance analysis, throughput, time-space tradeoff

14 Improving 64-Bit Java IPF Performance by Compressing Heap References Ali-Reza Adl-Tabatabai, Jay Bharadwaj, Michal Cierniak, Marsha Eng, Jesse Fang, Brian T. Lewis, Brian R. Murphy, James M. StichnothMarch 2004 **Proceedings of the international symposium on Code generation and optimization: feedback-directed and runtime optimization CGO '04**

Publisher: IEEE Computer Society

Full text available:  pdf(172.84 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

64-bit processor architectures like the Intel® Itanium®Processor Family are designed for large applicationsthat need large memory addresses. When runningapplications that fit within a 32-bit address space, 64-bitCPUs are at a disadvantage compared to 32-bit CPUsbecause of the larger memory footprints for their data. This results in worse cache and TLB utilization, and consequentlylower performance because of increased missratios. This paper considers software techniques for virtualmachines that all ...

15 Syncopation: generational real-time garbage collection in the metronome David F. Bacon, Perry Cheng, David Grove, Martin T. VechevJune 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems LCTES '05**, Volume 40 Issue 7

Publisher: ACM Press

Full text available:  pdf(212.34 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Real-time garbage collection has been shown to be feasible, but for programs with high allocation rates, the utilization achievable is not sufficient for some systems. Since a high allocation rate is often correlated with a more high-level, abstract programming style, the ability to provide good real-time performance for such programs will help continue to raise the level of abstraction at which real-time systems can be programmed. We have developed techniques that allow generational collection to ...

Keywords: allocation, garbage collection, real-time, scheduling

16 SAFECode: enforcing alias analysis for weakly typed languages Dinakar Dhurjati, Sumant Kowshik, Vikram AdveJune 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6

Publisher: ACM Press

Full text available:  pdf(216.22 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index](#)

[terms](#)

Static analysis of programs in weakly typed languages such as C and C++ is generally not sound because of possible memory errors due to dangling pointer references, uninitialized pointers, and array bounds overflow. We describe a compilation strategy for standard C programs that guarantees that aggressive interprocedural pointer analysis (or less precise ones), a call graph, and type information for a subset of memory, are never invalidated by any possible memory errors. We formalize our approach ...

Keywords: alias analysis, automatic pool allocation, compilers, programming languages, region management

17 Research papers: test & analysis II: Backwards-compatible array bounds checking

 [for C with very low overhead](#)

Dinakar Dhurjati, Vikram Adve

May 2006 **Proceeding of the 28th international conference on Software engineering ICSE '06**

Publisher: ACM Press

Full text available:  [pdf\(169.34 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

The problem of enforcing correct usage of array and pointer references in C and C++ programs remains unsolved. The approach proposed by Jones and Kelly (extended by Ruwase and Lam) is the only one we know of that does not require significant manual changes to programs, but it has extremely high overheads of 5x-6x and 11x-12x in the two versions. In this paper, we describe a collection of techniques that dramatically reduce the overhead of this approach, by exploiting a fine-grain partitioning of ...

Keywords: array bounds checking, automatic pool allocation, compilers, programming languages, region management

18 Gilgamesh: a multithreaded processor-in-memory architecture for petaflops computing

Thomas L. Sterling, Hans P. Zima

November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing Supercomputing '02**

Publisher: IEEE Computer Society Press

Full text available:  [pdf\(322.86 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Processor-in-Memory (PIM) architectures avoid the von Neumann bottleneck in conventional machines by integrating high-density DRAM and CMOS logic on the same chip. Parallel systems based on this new technology are expected to provide higher scalability, adaptability, robustness, fault tolerance and lower power consumption than current MPPs or commodity clusters. In this paper we describe the design of *Gilgamesh*, a PIM-based massively parallel architecture, and elements of its execution mo ...

Keywords: Petaflops computing, Processor-In-Memory, data parallel processing, irregular applications, parallel architectures

19 Masking the overhead of protocol layering

 [Robbert van Renesse](#)

August 1996 **ACM SIGCOMM Computer Communication Review , Conference proceedings on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '96**, Volume 26 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(149.53 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Protocol layering has been advocated as a way of dealing with the complexity of computer communication. It has also been criticized for its performance overhead. In this paper, we present some insights in the design of protocols, and how these insights can be used to mask the overhead of layering, in a way similar to client caching in a file system. With our techniques, we achieve an order of magnitude improvement in end-to-end message latency in the Horus communication framework. Over an ATM ne ...

20 IMCE: Integrated media creation environment

 Brett Adams, Svetha Venkatesh, Ramesh Jain

August 2005 **ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)**, Volume 1 Issue 3

Publisher: ACM Press

Full text available:  [pdf\(16.79 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We discuss the design goals for an integrated media creation environment (IMCE) aimed at enabling the average user to create media artifacts with professional qualities. The resulting requirements are implemented and we demonstrate the efficacy of the resulting system with the generation of two simple home movies. The significance for the average user seeking to create home movies lies in the flexible and automatic application of film principles to the task, removal of tedious low-level editing ...

Keywords: Home movie, computational media aesthetics, semantics, video collection

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
Search: The ACM Digital Library The Guide

cookie aligned addresses store metadata collect garbage

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used:

[cookie aligned addresses store metadata collect garbage](#)

Found 34,547 of 212,128

Sort results by [relevance](#)
 [Save results to a Binder](#)
[Try an Advanced Search](#)Display results [expanded form](#)
 [Search Tips](#)
[Try this search in The ACM Guide](#)
 [Open results in a new window](#)

Results 21 - 40 of 200

Result page: [previous](#)[1](#)[2](#)[3](#)[4](#)[5](#)[6](#)[7](#)[8](#)[9](#)[10](#)[next](#)

Best 200 shown

Relevance scale

**21** [Frangipani: a scalable distributed file system](#)

Chandramohan A. Thekkath, Timothy Mann, Edward K. Lee

 October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles SOSP '97**, Volume 31 Issue 5
Publisher: ACM PressFull text available: [pdf\(2.20 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**22** [Dynamic heap type inference for program understanding and debugging](#)

Marina Polishchuk, Ben Liblit, Chloë W. Schulze

 January 2007 **ACM SIGPLAN Notices , Proceedings of the 34th annual ACM SIGPLAN-SIGACT symposium on Principles of programming languages POPL '07**, Volume 42 Issue 1
Publisher: ACM PressFull text available: [pdf\(577.70 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

C programs can be difficult to debug due to lax type enforcement and low-level access to memory. We present a dynamic analysis for C that checks heap snapshots for consistency with program types. Our approach builds on ideas from physical subtyping and conservative garbage collection. We infer a program-defined type for each allocated storage location or identify "untypable" blocks that reveal heap corruption or type safety violations. The analysis exploits symbolic debug information if present, ...

Keywords: conservative garbage collection, constraints, debugging tools, dynamic type inference, heap visualization, physical subtyping

**23** [Cheap recovery: a key to self-managing state](#)

Andrew C. Huang, Armando Fox

 February 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 1
Publisher: ACM PressFull text available: [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Cluster hash tables (CHTs) are key components of many large-scale Internet services due to their highly-scalable performance and the prevalence of the type of data they store. Another advantage of CHTs is that they can be designed to be as self-managing as a cluster of stateless servers. One key to achieving this extreme manageability is reboot-based recovery that is predictably fast and has modest impact on system performance and availability. This "cheap" recovery mechanism simplifies manage...

Keywords: Cluster hash table, manageability, quorum replication, storage systems design

24 Application performance and flexibility on exokernel systems

 M. Frans Kaashoek, Dawson R. Engler, Gregory R. Ganger, Hector M. Briceño, Russell Hunt, David Mazières, Thomas Pinckney, Robert Grimm, John Jannotti, Kenneth Mackenzie
October 1997 **ACM SIGOPS Operating Systems Review, Proceedings of the sixteenth ACM symposium on Operating systems principles SOSP '97**, Volume 31 Issue 5

Publisher: ACM Press

Full text available:  [pdf\(2.39 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

25 Implementation issues: How to shadow every byte of memory used by a program

 Nicholas Nethercote, Julian Seward
June 2007 **Proceedings of the 3rd international conference on Virtual execution environments VEE '07**

Publisher: ACM Press

Full text available:  [pdf\(198.86 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Several existing dynamic binary analysis tools use *shadowmemory*-they shadow, in software, every byte of memory used by a program with another value that says something about it. Shadow memory is difficult to implement both efficiently and robustly. Nonetheless, existing shadow memory implementations have not been studied in detail. This is unfortunate, because shadow memory is powerful-for example, some of the existing tools that use it detect critical errors such as bad memory access...

Keywords: dynamic binary analysis, dynamic binary instrumentation, memcheck, shadow memory, valgrind

26 Applied cryptography: Attacking and repairing the winZip encryption scheme

 Tadayoshi Kohno
October 2004 **Proceedings of the 11th ACM conference on Computer and communications security CCS '04**

Publisher: ACM Press

Full text available:  [pdf\(171.91 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#), [review](#)

WinZip is a popular compression utility for Microsoft Windows computers, the latest version of which is advertised as having "easy-to-use AES encryption to protect your sensitive data." We exhibit several attacks against WinZip's new encryption method, dubbed "AE-2" or "Advanced Encryption, version two." We then discuss secure alternatives. Since at a high level the underlying WinZip encryption method appears secure (the core is exactly Encrypt-then-Authenticate using AES-CTR and HMAC-SHA1),

...

Keywords: WinZip, Zip, applied cryptography, attacks, compression, encryption, security fixes

27 Architectural Support for Software Transactional Memory

Bratin Saha, Ali-Reza Adl-Tabatabai, Quinn Jacobson
December 2006 **Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture MICRO 39**

Publisher: IEEE Computer Society

Full text available:  [pdf\(325.24 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Transactional memory provides a concurrency control mechanism that avoids many of the

pitfalls of lock-based synchronization. Researchers have proposed several different implementations of transactional memory, broadly classified into software transactional memory (STM) and hardware transactional memory (HTM). Both approaches have their pros and cons: STMs provide rich and flexible transactional semantics on stock processors but incur significant overheads. HTMs, on the other hand, provide high ...

28 Comprehensively and efficiently protecting the heap

 Mazen Kharbutli, Xiaowei Jiang, Yan Solihin, Guru Venkataramani, Milos Prvulovic
October 2006 **ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating Systems Review , Proceedings of the 12th international conference on Architectural support for programming languages and operating systems ASPLOS-XII**, Volume 41 , 34 , 40 Issue 11 , 5 , 5

Publisher: ACM Press

Full text available:  [pdf\(329.47 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The goal of this paper is to propose a scheme that provides comprehensive security protection for the heap. Heap vulnerabilities are increasingly being exploited for attacks on computer programs. In most implementations, the heap management library keeps the heap meta-data (heap structure information) and the application's heap data in an interleaved fashion and does not protect them against each other. Such implementations are inherently unsafe: vulnerabilities in the application can cause the ...

Keywords: computer security, heap attacks, heap security, heap server

29 Scalable parallel allocation: McRT-Malloc: a scalable transactional memory allocator

 Richard L. Hudson, Bratin Saha, Ali-Reza Adl-Tabatabai, Benjamin C. Hertzberg
June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available:  [pdf\(332.45 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Emerging multi-core processors promise to provide an exponentially increasing number of hardware threads with every generation. Applications will need to be highly concurrent to fully use the power of these processors. To enable maximum concurrency, libraries (such as malloc-free packages) would therefore need to use non-blocking algorithms. But lock-free algorithms are notoriously difficult to reason about and inappropriate for average programmers. Transactional memory promises to significantly ...

Keywords: memory management, runtimes, synchronization, transactional memory

30 Cost-effective object space management for hardware-assisted real-time garbage collection

 Kelvin D. Nilsen, William J. Schmidt
December 1992 **ACM Letters on Programming Languages and Systems (LOPLAS)**, Volume 1 Issue 4

Publisher: ACM Press

Full text available:  [pdf\(1.29 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern object-oriented languages and programming paradigms require finer-grain division of memory than is provided by traditional paging and segmentation systems. This paper describes the design of an OSM (Object Space Manager) that allows partitioning of real memory on object, rather than page, boundaries. The time required by the OSM to create an object, or to find the beginning of an object given a pointer to any location within it, is approximately one memory cycle. Object sizes are lim ...

Keywords: automatic garbage collection, dynamic storage management, high-level language architectures, memory technologies, real-time and embedded systems, run-time environments

31 Melange: creating a "functional" internet Anil Madhavapeddy, Alex Ho, Tim Deegan, David Scott, Ripduman SohanMarch 2007 **ACM SIGOPS Operating Systems Review , Proceedings of the 2007 conference on EuroSys EuroSys '07**, Volume 41 Issue 3**Publisher:** ACM PressFull text available:  [pdf\(760.25 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#)

Most implementations of critical Internet protocols are written in type-unsafe languages such as C or C++ and are regularly vulnerable to serious security and reliability problems. Type-safe languages eliminate many errors but are not used to due to the perceived performance overheads.

We combine two techniques to eliminate this performance penalty in a practical fashion: strong static typing and generative meta-programming. Static typing eliminates run-time type information by checkin ...

32 201 principles of software development Alan M. Davis

January 1995 Book

Publisher: McGraw-Hill, Inc.Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#), [review](#)**From the Preface**

If software engineering is really an engineering discipline, it is the intelligent application of proven principles, techniques, languages, and tools to the cost-effective creation and maintenance of software that satisfies users' needs. This book is the first collection of software engineering principles ever written in one volume. A principle is a basic truth, rule, or assumption about software engineering that holds regardless of the technique, tool, or ...

33 Facilitating software evolution research with kenyon Jennifer Bevan, E. James Whitehead, Sunghun Kim, Michael GodfreySeptember 2005 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 10th European software engineering conference held jointly with 13th ACM SIGSOFT international symposium on Foundations of software engineering ESEC/FSE-13**, Volume 30 Issue 5**Publisher:** ACM PressFull text available:  [pdf\(327.20 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Software evolution research inherently has several resource-intensive logistical constraints. Archived project artifacts, such as those found in source code repositories and bug tracking systems, are the principal source of input data. Analysis-specific facts, such as commit metadata or the location of design patterns within the code, must be extracted for each change or configuration of interest. The results of this resource-intensive "fact extraction" phase must be stored efficiently, for late ...

Keywords: software configuration management, software evolution, software stratigraphy

34 Cycles to recycle: garbage collection to the IA-64 Richard L. Hudson, J. Elliot Moss, Sreenivas Subramoney, Weldon WashburnOctober 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1**Publisher:** ACM PressFull text available:  [pdf\(1.25 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The IA-64, Intel's 64-bit instruction set architecture, exhibits a number of interesting architectural features. Here we consider those features as they relate to supporting garbage collection (GC). We aim to assist GC and compiler implementors by describing how one may exploit features of the IA-64. Along the way, we record some previously unpublished object scanning techniques, and offer novel ones for object allocation (suggesting some simple operating system support that would simplify it ...)

35 [Automatic pool allocation: improving performance by controlling data structure layout in the heap](#)

Chris Lattner, Vikram Adve

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05**, Volume 40

Issue 6

Publisher: ACM Press

Full text available: [pdf\(215.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper describes *Automatic Pool Allocation*, a transformation framework that segregates distinct instances of heap-based data structures into separate memory pools and allows heuristics to be used to partially control the internal layout of those data structures. The primary goal of this work is performance improvement, not automatic memory management, and the paper makes several new contributions. The key contribution is a new compiler algorithm for partitioning heap objects in impera ...

Keywords: cache, data layout, pool allocation, recursive data structure, static analysis

36 [Mondrix: memory isolation for linux using mondriaan memory protection](#)

Emmett Witchel, Junghwan Rhee, Krste Asanović

October 2005 **ACM SIGOPS Operating Systems Review , Proceedings of the twentieth ACM symposium on Operating systems principles SOSP '05**, Volume 39 Issue 5

Publisher: ACM Press

Full text available: [pdf\(332.09 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents the design and an evaluation of Mondrix, a version of the Linux kernel with Mondriaan Memory Protection (MMP). MMP is a combination of hardware and software that provides efficient fine-grained memory protection between multiple protection domains sharing a linear address space. Mondrix uses MMP to enforce isolation between kernel modules which helps detect bugs, limits their damage, and improves kernel robustness and maintainability. During development, MMP exposed two kernel ...

Keywords: fine-grained memory protection

37 [Efficient distributed backup with delta compression](#)

Randal C. Burns, Darrell D. E. Long

November 1997 **Proceedings of the fifth workshop on I/O in parallel and distributed systems IOPADS '97**

Publisher: ACM Press

Full text available: [pdf\(1.37 MB\)](#) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

38 [Heap analysis: Transparent pointer compression for linked data structures](#)

Chris Lattner, Vikram S. Adve

June 2005 **Proceedings of the 2005 workshop on Memory system performance MSP '05**

Publisher: ACM Press

Full text available: Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

[pdf\(377.13 KB\)](#)[terms](#)

64-bit address spaces are increasingly important for modern applications, but they come at a price: pointers use twice as much memory, reducing the effective cache capacity and memory bandwidth of the system (compared to 32-bit address spaces). This paper presents a sophisticated, automatic transformation that shrinks pointers from 64-bits to 32-bits. The approach is "macroscopic," i.e., it operates on an entire logical data structure in the program at a time. It allows an *individual* data ...

Keywords: cache, data layout, pointer compression, recursive data structure, static analysis

39 [Memory allocation: A locality-improving dynamic memory allocator](#)

 Yi Feng, Emery D. Berger

June 2005 **Proceedings of the 2005 workshop on Memory system performance MSP '05**

Publisher: ACM Press

Full text available: [pdf\(265.72 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

In general-purpose applications, most data is dynamically allocated. The memory manager therefore plays a crucial role in application performance by determining the spatial locality of heap objects. Previous general-purpose allocators have focused on reducing fragmentation, while most locality-improving allocators have either focused on improving the locality of the allocator (not the application), or required programmer hints or profiling to guide object placement. We present a high-performance ...

Keywords: allocator, cache locality, fragmentation, memory management, paging, vam, virtual memory

40 [Lightweight recoverable virtual memory](#)

 M. Satyanarayanan, Henry H. Mashburn, Puneet Kumar, David C. Steere, James J. Kistler
February 1994 **ACM Transactions on Computer Systems (TOCS)**, Volume 12 Issue 1

Publisher: ACM Press

Full text available: [pdf\(1.73 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Recoverable virtual memory refers to regions of a virtual address space on which transactional guarantees are offered. This article describes RVM, an efficient, portable, and easily used implementation of recoverable virtual memory for Unix environments. A unique characteristic of RVM is that it allows independent control over the transactional properties of atomicity, permanence, and serializability. This leads to considerable flexibility in the use of RVM, potentially enabling ...

Keywords: Camelot, Coda, RVM, Unix, logging, paging, persistence, scalability, throughput, truncation

Results 21 - 40 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
 The ACM Digital Library The Guide

cookie aligned addresses store metadata collect garbage

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used:

[cookie aligned addresses store metadata collect garbage](#)

Found 34,547 of 212,128

Sort results by relevance

 [Save results to a Binder](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Display results expanded form

 [Search Tips](#)
 [Open results in a new window](#)

Results 41 - 60 of 200

Result page: [previous](#)

1

2

3

4

5

6

7

8

9

10

[next](#)

Best 200 shown

Relevance scale

**41** [Age-based garbage collection](#)

Darko Stefanović, Kathryn S. McKinley, J. Eliot B. Moss

 October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available: [pdf\(1.47 MB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern generational garbage collectors look for garbage among the young objects, because they have high mortality; however, these objects include the very youngest objects, which clearly are still live. We introduce new garbage collection algorithms, called age-based, some of which postpone consideration of the youngest objects. Collecting less than the whole heap requires write barrier mechanisms to track pointers into the collected region. We describe her ...

Keywords: garbage collection, generational and copy collection, object behavior, write barrier

**42** [FAB: building distributed enterprise disk arrays from commodity components](#)

Yasushi Saito, Svend Frølund, Alistair Veitch, Arif Merchant, Susan Spence

 October 2004 **ACM SIGARCH Computer Architecture News , ACM SIGOPS Operating Systems Review , ACM SIGPLAN Notices , Proceedings of the 11th international conference on Architectural support for programming languages and operating systems ASPLOS-XI**, Volume 32 , 38 , 39 Issue 5 , 5 , 11

Publisher: ACM Press

Full text available: [pdf\(671.67 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This paper describes the design, implementation, and evaluation of a Federated Array of Bricks (FAB), a distributed disk array that provides the reliability of traditional enterprise arrays with lower cost and better scalability. FAB is built from a collection of *bricks*, small storage appliances containing commodity disks, CPU, NVRAM, and network interface cards. FAB deploys a new majority-voting-based algorithm to replicate or erasure-code logical blocks across bricks and a reconfigurati ...

Keywords: consensus, disk array, erasure coding, replication, storage, voting

**43** [A Self-Organizing Storage Cluster for Parallel Data-Intensive Applications](#)

Hong Tang, Aziz Gulbeden, Jingyu Zhou, William Strathearn, Tao Yang, Lingkun Chu

November 2004 **Proceedings of the 2004 ACM/IEEE conference on Supercomputing SC '04**

Publisher: IEEE Computer Society

Full text available: [pdf\(330.26 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Cluster-based storage systems are popular for data-intensive applications and it is desirable yet challenging to provide incremental expansion and high availability while achieving scalability and strong consistency. This paper presents the design and implementation of a self-organizing storage cluster called Sorrento, which targets data-intensive workload with highly parallel requests and low write-sharing patterns. Sorrento automatically adapts to storage node joins and departures, and the sys ...

44 Experiences with the Amoeba distributed operating system

 Andrew S. Tanenbaum, Robbert van Renesse, Hans van Staveren, Gregory J. Sharp, Sape J. Mullender

December 1990 **Communications of the ACM**, Volume 33 Issue 12

Publisher: ACM Press

Full text available: [pdf\(2.71 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Amoeba project is a research effort aimed at understanding how to connect multiple computers in a seamless way [16, 17, 26, 27, 31]. The basic idea is to provide the users with the illusion of a single powerful timesharing system, when, in fact, the system is implemented on a collection of machines, potentially distributed among several countries. This research has led to the design and implementation of the Amoeba distributed operating system, which is being used as a prototype and veh ...

45 The ExoVM system for automatic VM and application reduction

 Ben L. Titzer, Joshua Auerbach, David F. Bacon, Jens Palsberg

June 2007 **ACM SIGPLAN Notices , Proceedings of the 2007 ACM SIGPLAN conference on Programming language design and implementation PLDI '07**, Volume 42 Issue 6

Publisher: ACM Press

Full text available: [pdf\(510.60 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Embedded systems pose unique challenges to Java application developers and virtual machine designers. Chief among these challenges is the memory footprint of both the virtual machine and the applications that run within it. With the rapidly increasing set of features provided by the Java language, virtual machine designers are often forced to build custom implementations that make various tradeoffs between the footprint of the virtual machine and the subset of the Java language and class libr ...

Keywords: VM design, VM modularity, dead code elimination, embedded systems, feature analysis, persistence, pre-initialization, static analysis, static compilation

46 Mostly lock-free malloc

 Dave Dice, Alex Garthwaite

June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium on Memory management ISMM '02**, Volume 38 Issue 2 supplement

Publisher: ACM Press

Full text available: [pdf\(609.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Modern multithreaded applications, such as application servers and database engines, can severely stress the performance of user-level memory allocators like the ubiquitous malloc subsystem. Such allocators can prove to be a major scalability impediment for the applications that use them, particularly for applications with large numbers of threads running on high-order multiprocessor systems. This paper introduces Multi-Processor Restartable Critical Sections, or MP-RCS. MP-RCS permits user-level ...

Keywords: affinity, locality, lock-free operations, malloc, restartable critical sections

47 A formal framework for component deployment Yu David Liu, Scott F. SmithOctober 2006 **ACM SIGPLAN Notices , Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**, Volume 41 Issue 10**Publisher:** ACM PressFull text available:  pdf(592.54 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Software deployment is a complex process, and industrial-strength frameworks such as .NET, Java, and CORBA all provide explicit support for component deployment. However, these frameworks are not built around fundamental principles as much as they are engineering efforts closely tied to particulars of the respective systems. Here we aim to elucidate the fundamental principles of software deployment, in a platform-independent manner. Issues that need to be addressed include deployment unit design ...

Keywords: component, deployment, version**48 Tag-free garbage collection for strongly typed programming languages** Benjamin GoldbergMay 1991 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1991 conference on Programming language design and implementation PLDI '91**, Volume 26 Issue 6**Publisher:** ACM PressFull text available:  pdf(1.03 MB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)**49 BitVault: a highly reliable distributed data retention platform** Zheng Zhang, Qiao Lian, Shiding Lin, Wei Chen, Yu Chen, Chao JinApril 2007 **ACM SIGOPS Operating Systems Review**, Volume 41 Issue 2**Publisher:** ACM PressFull text available:  pdf(414.59 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper summarizes our experience designing and implementing BitVault: a content-addressable retention platform for large volumes of reference data -- seldom-changing information that needs to be retained for a long time. BitVault uses "smart bricks" as the building block to lower the hardware cost. The challenges are to keep management costs low in a system that scales from one brick to tens of thousands, to ensure reliability, and to deliver a simple design. Our design incorporates peer- ...

50 Compiler and runtime support for efficient software transactional memory Ali-Reza Adl-Tabatabai, Brian T. Lewis, Vijay Menon, Brian R. Murphy, Bratin Saha, Tatiana ShpeismanJune 2006 **ACM SIGPLAN Notices , Proceedings of the 2006 ACM SIGPLAN conference on Programming language design and implementation PLDI '06**, Volume 41 Issue 6**Publisher:** ACM PressFull text available:  pdf(211.55 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Programmers have traditionally used locks to synchronize concurrent access to shared data. Lock-based synchronization, however, has well-known pitfalls: using locks for fine-grain synchronization and composing code that already uses locks are both difficult and prone to deadlock. Transactional memory provides an alternate concurrency control mechanism that avoids these pitfalls and significantly eases concurrent programming. Transactional memory language constructs have recently been proposed as ...

Keywords: code generation, compiler optimizations, locking, synchronization,

transactional memory, virtual machines

51 New garbage collection algorithms and strategies: Dynamic selection of application-specific garbage collectors

 Sunil Soman, Chandra Krintz, David F. Bacon
October 2004 **Proceedings of the 4th international symposium on Memory management ISMM '04**

Publisher: ACM Press

Full text available:  pdf(185.74 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Much prior work has shown that the performance enabled by garbage collection (GC) systems is highly dependent upon the behavior of the application as well as on the available resources. That is, no single GC enables the best performance for all programs and all heap sizes. To address this limitation, we present the design, implementation, and empirical evaluation of a novel Java Virtual Machine (JVM) extension that facilitates dynamic switching between a number of very different and popular g ...

Keywords: Java, annotation, application-specific collection, dynamic selection, hot-swapping, virtual machine

52 Risks to the public: Risks to the public in computers and related systems

 Peter G. Neumann
May 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 3

Publisher: ACM Press

Full text available:  pdf(1.92 MB) Additional Information: [full citation](#)

53 XORs in the air: practical wireless network coding

 Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Médard, Jon Crowcroft
August 2006 **ACM SIGCOMM Computer Communication Review , Proceedings of the 2006 conference on Applications, technologies, architectures, and protocols for computer communications SIGCOMM '06**, Volume 36 Issue 4

Publisher: ACM Press

Full text available:  pdf(640.77 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index terms](#)

This paper proposes COPE, a new architecture for wireless mesh networks. In addition to forwarding packets, routers mix (i.e., code) packets from different sources to increase the information content of each transmission. We show that intelligently mixing packets increases network throughput. Our design is rooted in the theory of network coding. Prior work on network coding is mainly theoretical and focuses on multicast traffic. This paper aims to bridge theory with practice; it addresses the co ...

Keywords: network coding, wireless networks

54 Semantic web and web 2.0: The two cultures: mashing up web 2.0 and the semantic web

 Anupriya Ankolekar, Markus Krötzsch, Thanh Tran, Denny Vrandecic
May 2007 **Proceedings of the 16th international conference on World Wide Web WWW '07**

Publisher: ACM Press

Full text available:  pdf(686.71 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

A common perception is that there are two competing visions for the future evolution of the Web: the Semantic Web and Web 2.0. A closer look, though, reveals that the core technologies and concerns of these two approaches are complementary and that each

field can and must draw from the other's strengths. We believe that future web applications will retain the Web 2.0 focus on community and usability, while drawing on Semantic Web infrastructure to facilitate mashup-like information sharing. H ...

Keywords: RDF, blog, semantic web, vision, web 2.0

55 Concurrent garbage collection using hardware-assisted profiling 

 Timothy H. Heil, James E. Smith

October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available:  pdf(1.74 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

In the presence of on-chip multithreading, a Virtual Machine (VM) implementation can readily take advantage of *service threads* for enhancing performance by performing tasks such as profile collection and analysis, dynamic optimization, and garbage collection concurrently with program execution. In this context, a hardware-assisted profiling mechanism is proposed. The *Relational Profiling Architecture* (RPA) is designed from the top down. RPA is based on a relational model similar ...

56 Compact garbage collection tables 

 David Tarditi

October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available:  pdf(958.92 KB) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Garbage collection tables for finding pointers on the stack can be represented in 20-25% of the space previously reported. Live pointer information is often the same at many call sites because there are few pointers live across most call sites. This allows live pointer information to be represented compactly by a small index into a table of descriptions of pointer locations. The mapping from program counter values to those small indexes can be represented compactly using several techniques. T ...

57 Conservative garbage collection for general memory allocators 

 Gustavo Rodriguez-Rivera

October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available:  pdf(829.20 KB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

This paper explains a technique that integrates conservative garbage collection on top of general memory allocators. This is possible by using two data structures named malloc-tables and jump-tables that are computed at garbage collection time to map pointers to beginning of objects and their sizes. This paper describes malloc-tables and jump-tables, an implementation of a malloc/jump-table based conservative garbage collector for Doug Lea's memory allocator, and experimental results that com ...

Keywords: automatic memory management, conservative garbage collection, memory allocation

58 Transactional memory: McRT-STM: a high performance software transactional 

 memory system for a multi-core runtime

Bratin Saha, Ali-Reza Adl-Tabatabai, Richard L. Hudson, Chi Cao Minh, Benjamin Hertzberg March 2006 **Proceedings of the eleventh ACM SIGPLAN symposium on Principles and practice of parallel programming PPoPP '06**

Publisher: ACM Press

Full text available:  pdf(183.96 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#), [index](#)

terms

Applications need to become more concurrent to take advantage of the increased computational power provided by chip level multiprocessing. Programmers have traditionally managed this concurrency using locks (mutex based synchronization). Unfortunately, lock based synchronization often leads to deadlocks, makes fine-grained synchronization difficult, hinders composition of atomic primitives, and provides no support for error recovery. Transactions avoid many of these problems, and therefore, prom ...

Keywords: atomic constructs, runtime environment, software transactional memory, two-phase locking and read-versioning

59 Defending against an Internet-based attack on the physical world Simon Byers, Aviel D. Rubin, David KormannAugust 2004 **ACM Transactions on Internet Technology (TOIT)**, Volume 4 Issue 3**Publisher:** ACM PressFull text available:  pdf(863.61 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

We discuss the dangers that scalable Internet functionality may present to the real world, focusing upon an attack that is simple, yet can have great impact, which we believe may occur quite soon. We offer and critique various solutions to this class of attack and hope to provide a warning to the Internet community of what is currently possible. The attack is, to some degree, a consequence of the availability of private information on the Web, and the increase in the amount of personal informati ...

Keywords: Internet threats, automated attacks, cybercrime

60 Design of the Mneme persistent object store J. Eliot B. MossApril 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 2**Publisher:** ACM PressFull text available:  pdf(3.22 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

The Mneme project is an investigation of techniques for integrating programming language and database features to provide better support for cooperative, information-intensive tasks such as computer-aided software engineering. The project strategy is to implement efficient, distributed, persistent programming languages. We report here on the Mneme persistent object store, a fundamental component of the project, discussing its design and initial prototype. Mneme stores objects

Results 41 - 60 of 200

Result page: [previous](#) [1](#) [2](#) **3** [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  [Adobe Acrobat](#) [QuickTime](#) [Windows Media Player](#) [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
Search: The ACM Digital Library The Guide

cookie aligned addresses store metadata collect garbage

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used:

[cookie aligned addresses store metadata collect garbage](#)

Found 34,547 of 212,128

Sort results by relevance

 [Save results to a Binder](#)
[Try an Advanced Search](#)

Display results expanded form

 [Search Tips](#)
[Try this search in The ACM Guide](#)
 [Open results in a new window](#)

Results 61 - 80 of 200

Result page: [previous](#)

1

2

3

4

5

6

7

8

9

10

[next](#)

Best 200 shown

Relevance scale

**61** [Garbage collection on multiprocessors: Portable, mostly-concurrent, mostly-copying](#)
[garbage collection for multi-processors](#)

Antony L. Hosking

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available: [pdf\(196.04 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Modern commodity platforms increasingly support thread-level parallelism, which must be exploited by garbage collected applications. We describe the design and implementation of a portable *mostly-concurrent* mostly-copying garbage collector that exhibits scalable performance on multi-processors. We characterize its performance for heap-intensive workloads on two different multi-processor platforms, showing maximum pause times two orders of magnitude shorter than for fully stop-the-world co ...

Keywords: ambiguous-roots, concurrent, conservative, garbage collection, incremental, memory management, mostly-concurrent, mostly-copying, portability

**62** [An on-the-fly reference-counting garbage collector for java](#)
[Yossi Levianni, Erez Petrank](#)
January 2006 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 28 Issue 1

Publisher: ACM Press

Full text available: [pdf\(787.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Reference-counting is traditionally considered unsuitable for multiprocessor systems. According to conventional wisdom, the update of reference slots and reference-counts requires atomic or synchronized operations. In this work we demonstrate this is not the case by presenting a novel reference-counting algorithm suitable for a multiprocessor system that does not require any synchronized operation in its write barrier (not even a compare-and-swap type of synchronization). A second novelty of thi ...

Keywords: Programming languages, garbage collection, memory management, reference-counting

**63** [Space efficient conservative garbage collection](#)
[Hans-Juergen Boehm](#)
June 1993 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1993 conference on Programming language design and implementation PLDI '93**, Volume 28 Issue 6

Publisher: ACM Press

Full text available:  pdf(1.03 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We call a garbage collector conservative if it has only partial information about the location of pointers, and is thus forced to treat arbitrary bit patterns as though they might be pointers, in at least some cases. We show that some very inexpensive, but previously unused techniques can have dramatic impact on the effectiveness of conservative garbage collectors in reclaiming memory. Our most significant observation is that static data that appears to point to the heap should not result in ...

64 The Design of efficient initialization and crash recovery for log-based file systems

 over flash memory

Chin-Hsien Wu, Tei-Wei Kuo, Li-Pin Chang

November 2006 **ACM Transactions on Storage (TOS)**, Volume 2 Issue 4

Publisher: ACM Press

Full text available:  pdf(336.16 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

While flash memory has been widely adopted for storage systems for various embedded systems, issues of performance and reliability have started receiving growing attention in recent years. How to provide efficient roll back and quick mounting for flash-memory file systems has become an important research topic in recent years, in addition to the work on effective garbage collection and superb runtime performance. Such an observation motivates our work on the investigation of efficient initializa ...

Keywords: Flash memory, crash recovery, efficient initialization, embedded systems, file systems, storage systems

65 Defending against an Internet-based attack on the physical world

 Simon Byers, Aviel D. Rubin, David Kormann

November 2002 **Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society WPES '02**

Publisher: ACM Press

Full text available:  pdf(201.19 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We discuss the dangers that scalable Internet functionality may present to the real world, focusing on a simple yet impactful attack that we believe may occur quite soon. We offer and critique various solutions to this class of attack and hope to provide a warning to the Internet community of what is currently possible. The attack is, to some degree, a consequence of the availability of private information on the Web, and the increase in the amount of personal information that users must reveal ...

Keywords: Internet Threats, automated attacks, computer security, computer security, cybercrime, internet threats

66 Compiler construction: an advanced course

F. L. Bauer, F. L. De Remer, M. Griffiths, U. Hill, J. J. Horning, C. H. A. Koster, W. M.

McKeeman, P. C. Poole, W. M. Waite, G. Goos, J. Hartmanis

January 1974 Book

Publisher: Springer-Verlag New York, Inc.

Additional Information: [full citation](#), [abstract](#), [references](#), [cited by](#)

The Advanced Course took place from March 4 to 15, 1974 and was organized by the Mathematical Institute of the Technical University of Munich and the Leibniz Computing Center of the Bavarian Academy of Sciences, in co-operation with the European Communities, sponsored by the Ministry for Research and Technology of the Federal Republic of Germany and by the European Research Office, London.

67 Eliminating synchronization-related atomic operations with biased locking and bulkrebiasing

Kenneth Russell, David Detlefs

October 2006 **ACM SIGPLAN Notices , Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**, Volume 41 Issue 10

Publisher: ACM Press

Full text available: [pdf\(383.65 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Java™ programming language contains built-in synchronization primitives for use in constructing multithreaded programs. Efficient implementation of these synchronization primitives is necessary in order to achieve high performance. Recent research [9, 12, 10, 3, 7] has focused on the run-time elimination of the atomic operations required to implement object monitor synchronization primitives. This paper describes a novel technique called *store-free biased locking* which eliminates ...

Keywords: Java, atomic, bias, lock, monitor, optimization, rebias, reservation, revoke, synchronization

68 A unified theory of garbage collection

David F. Bacon, Perry Cheng, V. T. Rajan

October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10

Publisher: ACM Press

Full text available: [pdf\(223.52 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Tracing and reference counting are uniformly viewed as being fundamentally different approaches to garbage collection that possess very distinct performance properties. We have implemented high-performance collectors of both types, and in the process observed that the more we optimized them, the more similarly they behaved - that they seem to share some deep structure.

We present a formulation of the two algorithms that shows that they are in fact duals of each other. Intuitively, the ...

Keywords: graph algorithms, mark-and-sweep, reference counting, tracing

69 Older-first garbage collection in practice: evaluation in a Java Virtual Machine

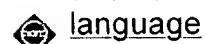
Darko Stefanović, Matthew Hertz, Stephen M. Blackburn, Kathryn S. McKinley, J. Eliot B. Moss

June 2002 **ACM SIGPLAN Notices , Proceedings of the 2002 workshop on Memory system performance MSP '02**, Volume 38 Issue 2 supplement

Publisher: ACM Press

Full text available: [pdf\(1.15 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#)

Until recently, the best performing copying garbage collectors used a generational policy which repeatedly collects the very youngest objects, copies any survivors to an older space, and then infrequently collects the older space. A previous study that used garbage-collection simulation pointed to potential improvements by using an *Older-First* copying garbage collection algorithm. The Older-First algorithm sweeps a fixed-sized window through the heap from older to younger objects, and avo ...

70 Protection traps and alternatives for memory management of an object-orientedlanguage

Antony L. Hosking, J. Eliot B. Moss

December 1993 **ACM SIGOPS Operating Systems Review , Proceedings of the**

**fourteenth ACM symposium on Operating systems principles SOSP
'93, Volume 27 Issue 5**

Publisher: ACM Press

Full text available:  [pdf\(1.48 MB\)](#)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Many operating systems allow user programs to specify the protection level (inaccessible, read-only, read-write) of pages in their virtual memory address space, and to handle any protection violations that may occur. Such page-protection techniques have been exploited by several user-level algorithms for applications including generational garbage collection and persistent stores. Unfortunately, modern hardware has made efficient handling of page protection faults more difficult. Moreover, page- ...

71 Content management: Dynamic program insertion in high quality video over IP 

 Taehyun Kim, Jack Brassil

June 2003 **Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video NOSSDAV '03**

Publisher: ACM Press

Full text available:  [pdf\(269.10 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We introduce an overlay network architecture and signaling mechanism that permit program insertions in live, high quality video streams transmitted over IP networks. We describe the implementation of an application proxy that dynamically inserts pre-recorded video programs into NTSC D1 quality Motion-JPEG streams with no visible artifacts. As increases in computing power further enable the modification of video during transport, new services such as personalized commercial advertisement insertio ...

Keywords: content delivery networks (CDNs), digital television (DTV), multimedia signaling, program and system information protocol (PSIP), program cues, real-time transport protocol (RTP), video streaming

72 Mark-copy: fast copying GC with less space overhead 

 Narendran Sachindran, J. Eliot, B. Moss

October 2003 **ACM SIGPLAN Notices , Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11

Publisher: ACM Press

Full text available:  [pdf\(297.93 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Copying garbage collectors have a number of advantages over non-copying collectors, including cheap allocation and avoiding fragmentation. However, in order to provide completeness (the guarantee to reclaim each garbage object eventually), standard copying collectors require space equal to twice the size of the maximum live data for a program. We present a *mark-copy* collection algorithm (MC) that extends generational copying collection and significantly reduces the heap space required to ...

Keywords: Java, copying collector, generational collector, mark-copy, mark-sweep

73 Accurate garbage collection in an uncooperative environment 

 Fergus Henderson

June 2002 **ACM SIGPLAN Notices , Proceedings of the 3rd international symposium on Memory management ISMM '02**, Volume 38 Issue 2 supplement

Publisher: ACM Press

Full text available:  [pdf\(136.65 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Previous attempts at garbage collection in uncooperative environments have generally used conservative or mostly-conservative approaches. We describe a technique for doing fully type-accurate garbage collection in an uncooperative environment, using a "shadow

stack" to link structs of pointer-containing variables, together with the data or code needed to trace them. We have implemented this in the Mercury compiler, which generates C code, and present preliminary performance data on the overheads ...

Keywords: C, garbage collection, multithreading, programming language implementation

74 Smalltalk-80: the language and its implementation

Adele Goldberg, David Robson
January 1983 Book

Publisher: Addison-Wesley Longman Publishing Co., Inc.

Full text available: [pdf\(33.56 MB\)](#) Additional Information: [full citation](#), [abstract](#), [cited by](#), [index terms](#), [review](#)

From the Preface (See Front Matter for full Preface)

Advances in the design and production of computer hardware have brought many more people into direct contact with computers. Similar advances in the design and production of computer software are required in order that this increased contact be as rewarding as possible. The Smalltalk-80 system is a result of a decade of research into creating computer software that is appropriate for producing highly functional and interactive ...

75 Ulterior reference counting: fast garbage collection without a long wait

 Stephen M. Blackburn, Kathryn S. McKinley
October 2003 **ACM SIGPLAN Notices, Proceedings of the 18th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '03**, Volume 38 Issue 11

Publisher: ACM Press

Full text available: [pdf\(218.61 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

General purpose garbage collectors have yet to combine short pause times with high throughput. For example, generational collectors can achieve high throughput. They have modest average pause times, but occasionally collect the whole heap and consequently incur long pauses. At the other extreme, concurrent collectors, including reference counting, attain short pause times but with significant performance penalties. This paper introduces a new hybrid collector that combines copying generational c ...

Keywords: Java, copying, generational hybrid, reference counting, ulterior reference counting

76 File and storage systems: Implementation of a Linux log-structured file system with a garbage collector

 Martin Jambor, Tomas Hruby, Jan Taus, Kuba Krchak, Vilim Holub
January 2007 **ACM SIGOPS Operating Systems Review**, Volume 41 Issue 1

Publisher: ACM Press

Full text available: [pdf\(477.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In many workloads, most write operations performed on a file system modify only a small number of blocks. The log-structured file system was designed for such a workload, additionally with the aim of fast crash recovery and system snapshots. Surprisingly, although implemented for Berkeley Sprite and BSD systems, there was no complete implementation for the current Linux kernel. In this paper, we present a complete implementation of the log-structured file system for the Linux kernel, which inclu ...

Keywords: Linux file systems, garbage collection, log-structured file systems

77 Efficient management for large-scale flash-memory storage systems with resource

conservation

 Li-Pin Chang, Tei-Wei Kuo
November 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 4

Publisher: ACM Press

Full text available:  pdf(1.45 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many existing approaches on flash-memory management are based on RAM-resident tables in which one single granularity size is used for both address translation and space management. As high-capacity flash memory is becoming more affordable than ever, the dilemma of how to manage the RAM space or how to improve the access performance is emerging for many vendors. In this article, we propose a tree-based management scheme which adopts multiple granularities in flash-memory management. Our objective ...

Keywords: Flash memory, consumer electronics, embedded systems, memory management, portable devices, storage systems

78 The measured cost of copying garbage collection mechanisms

 Michael W. Hicks, Jonathan T. Moore, Scott M. Nettles
August 1997 **ACM SIGPLAN Notices , Proceedings of the second ACM SIGPLAN international conference on Functional programming ICFP '97**, Volume 32 Issue 8

Publisher: ACM Press

Full text available:  pdf(1.65 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

We examine the costs and benefits of a variety of copying garbage collection (GC) mechanisms across multiple architectures and programming languages. Our study covers both low-level object representation and copying issues as well as the mechanisms needed to support more advanced techniques such as generational collection, large object spaces, and type segregated areas. Our experiments are made possible by a novel performance analysis tool, *Oscar*. Oscar allows us to capture snapshots of pr ...

79 Evolving a language in and for the real world: C++ 1991-2006

 Bjarne Stroustrup
June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programming languages HOPL III**

Publisher: ACM Press

Full text available:  pdf(838.10 KB) Additional Information: [full citation](#), [appendices and supplements](#), [abstract](#), [references](#), [index terms](#)

This paper outlines the history of the C++ programming language from the early days of its ISO standardization (1991), through the 1998 ISO standard, to the later stages of the C++0x revision of that standard (2006). The emphasis is on the ideals, constraints, programming techniques, and people that shaped the language, rather than the minutiae of language features. Among the major themes are the emergence of generic programming and the STL (the C++ standard library's algorithms and container ...

Keywords: C++, ISO, STL, evolution, history, language use, libraries, multi-paradigm programming, programming language, standardization

80 The case for profile-directed selection of garbage collectors

 Robert Fitzgerald, David Tarditi
October 2000 **ACM SIGPLAN Notices , Proceedings of the 2nd international symposium on Memory management ISMM '00**, Volume 36 Issue 1

Publisher: ACM Press

Full text available:  pdf(1.28 MB) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Many garbage-collected systems use a single garbage collection algorithm across all applications. It has long been known that this can produce poor performance on

applications for which that collector is not well suited. In some systems, such as those that execute stand-alone compiled executables, an appropriate collector for each application can be selected from a pool of available collectors and tuned by using profile information. In a study of 20 benchmarks and several collectors, compiled ...

Results 61 - 80 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) **4** [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.
[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
 The ACM Digital Library The Guide

cookie N-aligned addresses store metadata collect garbage

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used:

[cookie N aligned addresses store metadata collect garbage](#)

Found 30,971 of 212,128

Sort results by relevance [Save results to a Binder](#)Display results expanded form [Search Tips](#) [Open results in a new window](#)
[Try an Advanced Search](#)
[Try this search in The ACM Guide](#)

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

Best 200 shown

Relevance scale

1 [Cheap recovery: a key to self-managing state](#)

Andrew C. Huang, Armando Fox

February 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 1**Publisher:** ACM PressFull text available: [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Cluster hash tables (CHTs) are key components of many large-scale Internet services due to their highly-scalable performance and the prevalence of the type of data they store. Another advantage of CHTs is that they can be designed to be as self-managing as a cluster of stateless servers. One key to achieving this extreme manageability is reboot-based recovery that is predictably fast and has modest impact on system performance and availability. This "cheap" recovery mechanism simplifies manageme ...

Keywords: Cluster hash table, manageability, quorum replication, storage systems design

2 [Service-oriented and mobile computing: From representations to computations: the](#)
[evolution of web architectures](#)

Justin R. Erenkrantz, Michael Gorlick, Girish Suryanarayana, Richard N. Taylor

September 2007 **Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering ESEC-FSE '07****Publisher:** ACM PressFull text available: [pdf\(416.31 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

REpresentational State Transfer (REST) guided the creation and expansion of the modern web. What began as an internet-scale distributed hypermedia system is now a vast sea of shared and interdependent services. However, despite the expressive power of REST, not all of its benefits are consistently realized by working systems. To resolve the dissonance between the promise of REST and the reality of fielded systems, we critically examine numerous web architectures. Our investigation yields a se ...

Keywords: mobile code, network continuations, representational state transfer, web services

3 [Programming languages: Garbage collection for embedded systems](#)

David F. Bacon, Perry Cheng, David Grove

September 2004 **Proceedings of the 4th ACM international conference on Embedded software EMSOFT '04**

Publisher: ACM Press

Full text available: [pdf\(199.59 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Security concerns on embedded devices like cellular phones make Java an extremely attractive technology for providing third-party and user-downloadable functionality. However, garbage collectors have typically required several times the maximum live data set size (which is the minimum possible heap size) in order to run well. In addition, the size of the virtual machine (ROM) image and the size of the collector's data structures (metadata) have not been a concern for server- or workstation-orient ...

Keywords: compaction, fragmentation, mark-and-sweep, tracing

4 MC²: high-performance garbage collection for memory-constrained environments

 Narendran Sachindran, J. Eliot B. Moss, Emery D. Berger
October 2004 **ACM SIGPLAN Notices**, Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04, Volume 39 Issue 10

Publisher: ACM Press

Full text available: [pdf\(503.53 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Java is becoming an important platform for memory-constrained consumer devices such as PDAs and cellular phones, because it provides safety and portability. Since Java uses garbage collection, efficient garbage collectors that run in constrained memory are essential. Typical collection techniques used on these devices are mark-sweep and mark-compact. Mark-sweep collectors can provide good throughput and pause times but suffer from fragmentation. Mark-compact collectors prevent fragmentation, ...

Keywords: copying collector, generational collector, java, mark-compact, mark-copy, mark-sweep, memory-constrained copying

5 Implementation issues: How to shadow every byte of memory used by a program

 Nicholas Nethercote, Julian Seward
June 2007 **Proceedings of the 3rd international conference on Virtual execution environments VEE '07**

Publisher: ACM Press

Full text available: [pdf\(198.86 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Several existing dynamic binary analysis tools use *shadowmemory*-they shadow, in software, every byte of memory used by a program with another value that says something about it. Shadow memory is difficult to implement both efficiently and robustly. Nonetheless, existing shadow memory implementations have not been studied in detail. This is unfortunate, because shadow memory is powerful-for example, some of the existing tools that use it detect critical errors such as bad memory access ...

Keywords: dynamic binary analysis, dynamic binary instrumentation, memcheck, shadow memory, valgrind

6 Risks to the public: Risks to the public in computers and related systems

 Peter G. Neumann
May 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 3

Publisher: ACM Press

Full text available: [pdf\(1.92 MB\)](#) Additional Information: [full citation](#)

7 Garbage collection: Reducing generational copy reserve overhead with fallback

 **compaction**

Phil McGachey, Antony L. Hosking

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06**

Publisher: ACM Press

Full text available:  [pdf\(817.15 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

As programming languages with managed runtimes become increasingly popular, it is essential that virtual machines are implemented efficiently. The performance of the memory management subsystem can be a defining factor in the performance of the virtual machine as a whole. We present a technique by which garbage collector performance can be improved. We describe an algorithm that combines a standard generational copying collector with a mark and compact collector. We observe that, since most object ...

Keywords: copying collector, garbage collection, generational collector, java, mark and compact

8 Garbage collection without paging 

 Matthew Hertz, Yi Feng, Emery D. Berger

June 2005 **ACM SIGPLAN Notices , Proceedings of the 2005 ACM SIGPLAN conference on Programming language design and implementation PLDI '05**, Volume 40

Issue 6

Publisher: ACM Press

Full text available:  [pdf\(231.14 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Garbage collection offers numerous software engineering advantages, but interacts poorly with virtual memory managers. Existing garbage collectors require far more pages than the application's working set and touch pages without regard to which ones are in memory, especially during full-heap garbage collection. The resulting paging can cause throughput to plummet and pause times to spike up to seconds or even minutes. We present a garbage collector that avoids paging. This *bookmarking collect* ...

Keywords: bookmarking collection, garbage collection, generational collection, memory pressure, paging, virtual memory

9 Virgil: objects on the head of a pin 

 Ben L. Titzer

October 2006 **ACM SIGPLAN Notices , Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications OOPSLA '06**, Volume 41 Issue 10

Publisher: ACM Press

Full text available:  [pdf\(487.18 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Embedded microcontrollers are becoming increasingly prolific, serving as the primary or auxiliary processor in products and research systems from microwaves to sensor networks. Microcontrollers represent perhaps the most severely resource-constrained embedded processors, often with as little as a few bytes of memory and a few kilobytes of code space. Language and compiler technology has so far been unable to bring the benefits of modern object-oriented languages to such processors. In this paper ...

Keywords: data-sensitive optimization, dead code elimination, embedded systems, heap compression, microcontrollers, multi-stage computation, sensor networks, standalone programs, static analysis, systems software, whole-program compilation

10

The Design of efficient initialization and crash recovery for log-based file systems over flash memory 

 Chin-Hsien Wu, Tei-Wei Kuo, Li-Pin Chang
November 2006 **ACM Transactions on Storage (TOS)**, Volume 2 Issue 4

Publisher: ACM Press

Full text available:  pdf(336.16 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

While flash memory has been widely adopted for storage systems for various embedded systems, issues of performance and reliability have started receiving growing attention in recent years. How to provide efficient roll back and quick mounting for flash-memory file systems has become an important research topic in recent years, in addition to the work on effective garbage collection and superb runtime performance. Such an observation motivates our work on the investigation of efficient initializa ...

Keywords: Flash memory, crash recovery, efficient initialization, embedded systems, file systems, storage systems

11 A unified theory of garbage collection

 David F. Bacon, Perry Cheng, V. T. Rajan
October 2004 **ACM SIGPLAN Notices , Proceedings of the 19th annual ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '04**, Volume 39 Issue 10

Publisher: ACM Press

Full text available:  pdf(223.52 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Tracing and reference counting are uniformly viewed as being fundamentally different approaches to garbage collection that possess very distinct performance properties. We have implemented high-performance collectors of both types, and in the process observed that the more we optimized them, the more similarly they behaved - that they seem to share some deep structure.

We present a formulation of the two algorithms that shows that they are in fact duals of each other. Intuitively, the ...

Keywords: graph algorithms, mark-and-sweep, reference counting, tracing

12 Mostly-copying reachability-based orthogonal persistence

 Antony L. Hosking, Jiawan Chen
October 1999 **ACM SIGPLAN Notices , Proceedings of the 14th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications OOPSLA '99**, Volume 34 Issue 10

Publisher: ACM Press

Full text available:  pdf(3.25 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

We describe how reachability-based orthogonal persistence can be supported even in uncooperative implementations of languages such as C++ and Modula-3, and without modification to the compiler. Our scheme extends Bartlett's mostly-copying garbage collector to manage both transient objects and resident persistent objects, and to compute the reachability closure necessary for stabilization of the persistent heap. It has been implemented in our prototype of reachability-based persistence for M ...

13 CCured: type-safe retrofitting of legacy software

 George C. Necula, Jeremy Condit, Matthew Harren, Scott McPeak, Westley Weimer
May 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 3

Publisher: ACM Press

Full text available:  pdf(613.66 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

This article describes CCured, a program transformation system that adds type safety guarantees to existing C programs. CCured attempts to verify statically that memory

errors cannot occur, and it inserts run-time checks where static verification is insufficient. CCured extends C's type system by separating pointer types according to their usage, and it uses a surprisingly simple type inference algorithm that is able to infer the appropriate pointer kinds for existing C programs. CCured uses phys ...

Keywords: Memory safety, libraries, pointer qualifier, subtyping

14 [File and storage systems: Implementation of a Linux log-structured file system with a garbage collector](#)



Martin Jambor, Tomas Hruba, Jan Taus, Kuba Krchak, Viliam Holub

January 2007 **ACM SIGOPS Operating Systems Review**, Volume 41 Issue 1

Publisher: ACM Press

Full text available: [pdf\(477.02 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In many workloads, most write operations performed on a file system modify only a small number of blocks. The log-structured file system was designed for such a workload, additionally with the aim of fast crash recovery and system snapshots. Surprisingly, although implemented for Berkeley Sprite and BSD systems, there was no complete implementation for the current Linux kernel. In this paper, we present a complete implementation of the log-structured file system for the Linux kernel, which incl ...

Keywords: Linux file systems, garbage collection, log-structured file systems

15 [Efficient management for large-scale flash-memory storage systems with resource conservation](#)



Li-Pin Chang, Tei-Wei Kuo

November 2005 **ACM Transactions on Storage (TOS)**, Volume 1 Issue 4

Publisher: ACM Press

Full text available: [pdf\(1.45 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Many existing approaches on flash-memory management are based on RAM-resident tables in which one single granularity size is used for both address translation and space management. As high-capacity flash memory is becoming more affordable than ever, the dilemma of how to manage the RAM space or how to improve the access performance is emerging for many vendors. In this article, we propose a tree-based management scheme which adopts multiple granularities in flash-memory management. Our objective ...

Keywords: Flash memory, consumer electronics, embedded systems, memory management, portable devices, storage systems

16 [Evolving a language in and for the real world: C++ 1991-2006](#)



Bjarne Stroustrup

June 2007 **Proceedings of the third ACM SIGPLAN conference on History of programming languages HOPL III**

Publisher: ACM Press

Full text available: [pdf\(838.10 KB\)](#) Additional Information: [full citation](#), [appendices and supplements](#), [abstract](#), [references](#), [index terms](#)

This paper outlines the history of the C++ programming language from the early days of its ISO standardization (1991), through the 1998 ISO standard, to the later stages of the C++0x revision of that standard (2006). The emphasis is on the ideals, constraints, programming techniques, and people that shaped the language, rather than the minutiae of language features. Among the major themes are the emergence of generic programming and the STL (the C++ standard library's algorithms and container ...

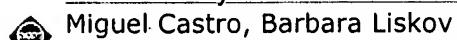
Keywords: C++, ISO, STL, evolution, history, language use, libraries, multi-paradigm programming, programming language, standardization

17 hFS: a hybrid file system prototype for improving small file and metadata

Zhihui Zhang, Kanad Ghose

March 2007 **ACM SIGOPS Operating Systems Review , Proceedings of the 2007 conference on EuroSys EuroSys '07**, Volume 41 Issue 3**Publisher:** ACM PressFull text available: [pdf\(429.39 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Two oft-cited file systems, the Fast File System (FFS) and the Log-Structured File System (LFS), adopt two sharply different update strategies---*update-in-place* and *update-out-of-place*. This paper introduces the design and implementation of a hybrid file system called hFS, which combines the strengths of FFS and LFS while avoiding their weaknesses. This is accomplished by distributing file system data into *two partitions* based on their size and type. In hFS, data blocks o ...

Keywords: disk inodes, file systems, metadata journaling, update strageties**18 Practical byzantine fault tolerance and proactive recovery**

Miguel Castro, Barbara Liskov

November 2002 **ACM Transactions on Computer Systems (TOCS)**, Volume 20 Issue 4**Publisher:** ACM PressFull text available: [pdf\(1.63 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

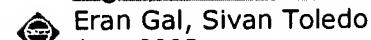
Our growing reliance on online services accessible on the Internet demands highly available systems that provide correct service without interruptions. Software bugs, operator mistakes, and malicious attacks are a major cause of service interruptions and they can cause arbitrary behavior, that is, Byzantine faults. This article describes a new replication algorithm, BFT, that can be used to build highly available systems that tolerate Byzantine faults. BFT can be used in practice to implement re ...

Keywords: Byzantine fault tolerance, asynchronous systems, proactive recovery, state machine replication, state transfer**19 Software support for portable storage: A superblock-based flash translation layer for****NAND flash memory**

Jeong-Uk Kang, Heeseung Jo, Jin-Soo Kim, Joonwon Lee

October 2006 **Proceedings of the 6th ACM & IEEE International conference on Embedded software EMSOFT '06****Publisher:** ACM PressFull text available: [pdf\(487.09 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In NAND flash-based storage systems, an intermediate software layer called a flash translation layer (FTL) is usually employed to hide the erase-before-write characteristics of NAND flash memory. This paper proposes a novel superblock-based FTL scheme, which combines a set of adjacent logical blocks into a superblock. In the proposed FTL scheme, superblocks are mapped at coarse granularity, while pages inside the superblock are mapped freely at fine granularity to any location in several physical b ...

Keywords: NAND flash memory, address translation, flash translation layer (FTL)**20 Algorithms and data structures for flash memories**

Eran Gal, Sivan Toledo

June 2005 **ACM Computing Surveys (CSUR)**, Volume 37 Issue 2**Publisher:** ACM Press

Full text available: [!\[\]\(c386b6be28f29f50a8089f4302e31f17_img.jpg\) pdf\(343.39 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Flash memory is a type of electrically-erasable programmable read-only memory (EEPROM). Because flash memories are nonvolatile and relatively dense, they are now used to store files and other persistent objects in handheld computers, mobile phones, digital cameras, portable music players, and many other computer systems in which magnetic disks are inappropriate. Flash, like earlier EEPROM devices, suffers from two limitations. First, bits can only be cleared by erasing a large block of memory. S ...

Keywords: EEPROM memory, Flash memory, wear leveling

Results 1 - 20 of 200

Result page: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads: [!\[\]\(3b6550c266b425c643039d03ba687257_img.jpg\) Adobe Acrobat](#) [!\[\]\(cd0c71aff64d61c958bc9f11c4e5892a_img.jpg\) QuickTime](#) [!\[\]\(36d2bc81059071ce463355c7cf7ed718_img.jpg\) Windows Media Player](#) [!\[\]\(0a085befc296d7093713b11162a43db5_img.jpg\) Real Player](#)


[Subscribe \(Full Service\)](#) [Register \(Limited Service, Free\)](#) [Login](#)
 The ACM Digital Library The Guide

cookie N-aligned addresses store metadata collect garbage

THE ACM DIGITAL LIBRARY

[Feedback](#) [Report a problem](#) [Satisfaction survey](#)

Terms used:

[cookie N aligned addresses store metadata collect garbage](#)

Found 30,971 of 212,128

Sort results
by

relevance

[Save results to a Binder](#)Display
results

expanded form

[Search Tips](#)
 [Open results in a new window](#)
[Try an Advanced Search](#)[Try this search in The ACM Guide](#)

Results 21 - 40 of 200

Result page: [previous](#)

1

2

3

4

5

6

7

8

9

10

[next](#)

Best 200 shown

Relevance scale

**21** [Services: ELF: an efficient log-structured flash file system for micro sensor nodes](#)

Hui Dai, Michael Neufeld, Richard Han

November 2004 **Proceedings of the 2nd international conference on Embedded networked sensor systems SenSys '04**

Publisher: ACM Press

Full text available: [pdf\(291.68 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

An efficient and reliable file storage system is important to micro sensor nodes so that data can be logged for later asynchronous delivery across a multi-hop wireless sensor network. Designing and implementing such a file system for a sensor node faces various challenges. Sensor nodes are highly resource constrained in terms of limited runtime memory, limited persistent storage, and finite energy. Also, the flash storage medium on sensor nodes differs in a variety of ways from the traditional ...

Keywords: eeprom, file system, flash, log structured, reliability, sensor**22** [An efficient and backwards-compatible transformation to ensure memory safety of C programs](#)

Wei Xu, Daniel C. DuVarney, R. Sekar

October 2004 **ACM SIGSOFT Software Engineering Notes , Proceedings of the 12th ACM SIGSOFT twelfth international symposium on Foundations of software engineering SIGSOFT '04/FSE-12, Volume 29 Issue 6**

Publisher: ACM Press

Full text available: [pdf\(115.69 KB\)](#)Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Memory-related errors, such as buffer overflows and dangling pointers, remain one of the principal reasons for failures of C programs. As a result, a number of recent research efforts have focused on the problem of dynamic detection of memory errors in C programs. However, existing approaches suffer from one or more of the following problems: inability to detect all memory errors (e.g., Purify), requiring non-trivial modifications to existing C programs (e.g., Cyclone), changing the memory ma ...

Keywords: C, memory safety, program transformation**23** [Real-time garbage collection for flash-memory storage systems of real-time embedded systems](#)

Li-Pin Chang, Tei-Wei Kuo, Shi-Wu Lo

November 2004 **ACM Transactions on Embedded Computing Systems (TECS)**, Volume 3

Issue 4

Publisher: ACM PressFull text available:  [pdf\(465.38 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Flash-memory technology is becoming critical in building embedded systems applications because of its shock-resistant, power economic, and nonvolatile nature. With the recent technology breakthroughs in both capacity and reliability, flash-memory storage systems are now very popular in many types of embedded systems. However, because flash memory is a write-once and bulk-erase medium, we need a translation layer and a garbage-collection mechanism to provide applications a transparent storage ...

Keywords: Embedded systems, flash memory, garbage collection, real-time system, storage systems

24 Faster high-level language virtual machines: Automatic feedback-directed object 

 [inlining in the java hotspot™ virtual machine](#)

Christian Wimmer, Hanspeter Mössenböck

June 2007 **Proceedings of the 3rd international conference on Virtual execution environments VEE '07****Publisher:** ACM PressFull text available:  [pdf\(341.49 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Object inlining is an optimization that embeds certain referenced objects into their referencing object. It reduces the costs of field accesses by eliminating unnecessary field loads. The order of objects in the heap is changed in such a way that objects that are accessed together are placed next to each other in memory so that their offset is fixed, i.e. the objects are *colocated*. This allows field loads to be replaced by address arithmetic. We implemented this optimization for ...

Keywords: cache, garbage collection, java, just-in-time compilation, object colocation, object inlining, optimization, performance

25 The Conquest file system: Better performance through a disk/persistent-RAM hybrid 

 [design](#)

An-I Andy Wang, Geoff Kuenning, Peter Reiher, Gerald Popek

August 2006 **ACM Transactions on Storage (TOS)**, Volume 2 Issue 3**Publisher:** ACM PressFull text available:  [pdf\(1.34 MB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Modern file systems assume the use of disk, a system-wide performance bottleneck for over a decade. Current disk caching and RAM file systems either impose high overhead to access memory content or fail to provide mechanisms to achieve data persistence across reboots. The *Conquest* file system is based on the observation that memory is becoming inexpensive, which enables all file system services to be delivered from memory, except for providing large storage capacity. Unlike caching, *Con* ...

Keywords: Persistent RAM, file systems, performance measurement, storage management

26 The KaffeOS Java runtime system 

 Godmar Back, Wilson C. Hsieh

July 2005 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 27 Issue 4**Publisher:** ACM PressFull text available:  [pdf\(704.30 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#), [review](#)

Single-language runtime systems, in the form of Java virtual machines, are widely deployed platforms for executing untrusted mobile code. These runtimes provide some of the features that operating systems provide: interapplication memory protection and basic system services. They do not, however, provide the ability to isolate applications from each other. Neither do they provide the ability to limit the resource consumption of applications. Consequently, the performance of current systems degra ...

Keywords: Robustness, garbage collection, isolation, language runtimes, resource management, termination, virtual machines

27 Quantifying the performance of garbage collection vs. explicit memory management

 Matthew Hertz, Emery D. Berger

October 2005 **ACM SIGPLAN Notices , Proceedings of the 20th annual ACM SIGPLAN conference on Object oriented programming, systems, languages, and applications OOPSLA '05**, Volume 40 Issue 10

Publisher: ACM Press

Full text available: .pdf(1.51 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Garbage collection yields numerous software engineering benefits, but its quantitative impact on performance remains elusive. One can compare the cost of *conservative* garbage collection to explicit memory management in C/C++ programs by linking in an appropriate collector. This kind of direct comparison is not possible for languages designed for garbage collection (e.g., Java), because programs in these languages naturally do not contain calls to free. Thus, the actual gap between the tim ...

Keywords: explicit memory management, garbage collection, oracular memory management, paging, performance analysis, throughput, time-space tradeoff

28 The Google file system

 Sanjay Ghemawat, Howard Gobioff, Shun-Tak Leung

October 2003 **ACM SIGOPS Operating Systems Review , Proceedings of the nineteenth ACM symposium on Operating systems principles SOSP '03**, Volume 37 Issue 5

Publisher: ACM Press

Full text available: .pdf(275.54 KB) Additional Information: [full citation](#), [references](#), [citations](#), [index terms](#)

Keywords: clustered storage, data storage, fault tolerance, scalability

29 Power to the people: end-user building of digital library collections

 Ian H. Witten, David Bainbridge, Stefan J. Boddie

January 2001 **Proceedings of the 1st ACM/IEEE-CS joint conference on Digital libraries JCDL '01**

Publisher: ACM Press

Full text available: .pdf(402.80 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Naturally, digital library systems focus principally on the reader: the consumer of the material that constitutes the library. In contrast, this paper describes an interface that makes it easy for people to build their own library collections. Collections may be built and served locally from the user's own web server, or (given appropriate permissions) remotely on a shared digital library host. End users can easily build new collections styled after existing ones from material on the Web o ...

30 Syncopation: generational real-time garbage collection in the metronome

David F. Bacon, Perry Cheng, David Grove, Martin T. Vechev



June 2005 **ACM SIGPLAN Notices** , Proceedings of the 2005 ACM SIGPLAN/SIGBED conference on Languages, compilers, and tools for embedded systems **LCTES '05**, Volume 40 Issue 7

Publisher: ACM Press

Full text available: [pdf\(212.34 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Real-time garbage collection has been shown to be feasible, but for programs with high allocation rates, the utilization achievable is not sufficient for some systems. Since a high allocation rate is often correlated with a more high-level, abstract programming style, the ability to provide good real-time performance for such programs will help continue to raise the level of abstraction at which real-time systems can be programmed. We have developed techniques that allow generational collection to ...

Keywords: allocation, garbage collection, real-time, scheduling

31 Industrial sessions: beyond relational tables: Coordinating backup/recovery and data consistency between database and file systems



Suparna Bhattacharya, C. Mohan, Karen W. Brannon, Inderpal Narang, Hui-I Hsiao, Mahadevan Subramanian

June 2002 **Proceedings of the 2002 ACM SIGMOD international conference on Management of data SIGMOD '02**

Publisher: ACM Press

Full text available: pdf(1.44 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Managing a combined store consisting of database data and file data in a robust and consistent manner is a challenge for database systems and content management systems. In such a hybrid system, images, videos, engineering drawings, etc. are stored as files on a file server while meta-data referencing/indexing such files is created and stored in a relational database to take advantage of efficient search. In this paper we describe solutions for two potentially problematic aspects of such a data ...

Keywords: DB2, content management, database backup, database recovery, datalinks

32 Designing a trace format for heap allocation events



Trishul Chilimbi, Richard Jones, Benjamin Zorn

October 2000 **ACM SIGPLAN Notices**, Proceedings of the 2nd international symposium on Memory management ISMM '00, Volume 36 Issue 1

Publisher: ACM Press

Full text available: [pdf \(1.53 MB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

Dynamic storage allocation continues to play an important role in the performance and correctness of systems ranging from user productivity software to high-performance servers. While algorithms for dynamic storage allocation have been studied for decades, much of the literature is based on measuring the performance of benchmark programs unrepresentative of many important allocation-intensive workloads. Furthermore, to date no standard has emerged or been proposed for publishing and exchange ...

33 Imposing a Memory Management Discipline on Software Deployment

Eelco Dolstra, Eelco Visser, Merijn de Jonge

May 2004 **Proceedings of the 26th International Conference on Software Engineering ICSE '04**

Publisher: IEEE Computer Society

Full text available: [pdf \(365.54 KB\)](#) Additional Information: [full citation](#), [abstract](#), [citations](#), [index terms](#)

The deployment of software components frequently fails because dependencies on other components are not declared explicitly or are declared imprecisely. This results in an incomplete reproduction of the environment necessary for proper operation, or in interference between incompatible variants. In this paper we show that these deployment hazards are similar to pointer hazards in memory models of programming

languages and can be countered by imposing a memory management discipline on software deployment ...

34 Joeq: a virtual machine and compiler infrastructure
John Whaley
June 2003 **Proceedings of the 2003 workshop on Interpreters, virtual machines and
emulators IVME '03**
Publisher: ACM Press

Joeq is a virtual machine and compiler infrastructure designed to facilitate research in virtual machine technologies such as Just-In-Time and Ahead-Of-Time compilation, advanced garbage collection techniques, distributed computation, sophisticated scheduling algorithms, and advanced run time techniques. Joeq is entirely implemented in Java, leading to reliability, portability, maintainability, and efficiency. It is also language-independent, so code from any supported language can be seamlessly ...

35 SAFECode: enforcing alias analysis for weakly typed languages
Dinakar Dhurjati, Sumant Kowshik, Vikram Adve
June 2006 **ACM SIGPLAN Notices** , Proceedings of the 2006 ACM SIGPLAN conference
on Programming language design and implementation **PLDI '06**, Volume 41
Issue 6
Publisher: ACM Press

Static analysis of programs in weakly typed languages such as C and C++ is generally not sound because of possible memory errors due to dangling pointer references, uninitialized pointers, and array bounds overflow. We describe a compilation strategy for standard C programs that guarantees that aggressive interprocedural pointer analysis (or less precise ones), a call graph, and type information for a subset of memory, are never invalidated by any possible memory errors. We formalize our approach ...

Keywords: alias analysis, automatic pool allocation, compilers, programming languages, region management

36 Web mining for web personalization
Magdalini Eirinaki, Michalis Vazirgiannis
February 2003 **ACM Transactions on Internet Technology (TOIT)**, Volume 3 Issue 1
Publisher: ACM Press

Web personalization is the process of customizing a Web site to the needs of specific users, taking advantage of the knowledge acquired from the analysis of the user's navigational behavior (usage data) in correlation with other information collected in the Web context, namely, structure, content, and user profile data. Due to the explosive growth of the Web, the domain of Web personalization has gained great momentum both in the research and commercial areas. In this article we present a survey ...

Keywords: WWW, Web personalization, Web usage mining, user profiling

37 Workshop on compositional software architectures: workshop report
May 1998 **ACM SIGSOFT Software Engineering Notes**, Volume 23 Issue 3
Publisher: ACM Press
Full text available:  pdf(2.91 MB) Additional Information: [full citation](#), [index terms](#)

38 Garbage collection on multiprocessors: Improving locality with parallel hierarchical **copying GC**

David Siegwart, Martin Hirzel

June 2006 **Proceedings of the 2006 international symposium on Memory management ISMM '06****Publisher:** ACM PressFull text available:  [pdf\(284.50 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper shows how to reduce cache and TLB misses by changing the order in which a parallel garbage collector copies heap objects. Reducing cache and TLB misses improves program run time. Parallel garbage collection improves scaling on multi-processor machines. Technology trends indicate that both memory locality and multi-processor scaling increase in importance. Our new algorithm is based on the earlier single-threaded "hierarchical decomposition" algorithm by Wilson, Lam, and Moher. This pa ...

Keywords: cache locality, generational, parallel**39 Singularity: rethinking the software stack** Galen C. Hunt, James R. LarusApril 2007 **ACM SIGOPS Operating Systems Review**, Volume 41 Issue 2**Publisher:** ACM PressFull text available:  [pdf\(428.24 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

Every operating system embodies a collection of design decisions. Many of the decisions behind today's most popular operating systems have remained unchanged, even as hardware and software have evolved. Operating systems form the foundation of almost every software stack, so inadequacies in present systems have a pervasive impact. This paper describes the efforts of the Singularity project to re-examine these design choices in light of advances in programming languages and verification tools. ...

Keywords: hardware protection domains, manifest-based programs (MBPs), operating systems, program specification, program verification, safe programming languages, sealed kernel, sealed process architecture, software-isolated processes (SIPs), unsafe code tax**40 Charles W. Bachman interview: September 25-26, 2004; Tucson, Arizona** Thomas HaighJanuary 2006 **ACM Oral History interviews****Publisher:** ACM PressFull text available:  [pdf\(761.66 KB\)](#) Additional Information: [full citation](#), [abstract](#)

Charles W. Bachman reviews his career. Born during 1924 in Kansas, Bachman attended high school in East Lansing, Michigan before joining the Army Anti Aircraft Artillery Corp, with which he spent two years in the Southwest Pacific Theater, during World War II.

After his discharge from the military, Bachman earned a B.Sc. in Mechanical Engineering in 1948, followed immediately by an M.Sc. in the same discipline, from the University of Pennsylvania. On graduation, he went to work for Do ...

Results 21 - 40 of 200

Result page: [previous](#) [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2007 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)